rchitektury Klient-Serwer

miesięcznik profesjonalistów

6
1997

# informatyka

Cena: 5,5 zł. ISSN 0542 9951 WYDAWNICTWO SIGMA NOT

P/1877/97

Pismo informatyczne - ukazuje się od 1965 roku

## Fatamorgana, mechanizmy napędowe i korzenie informatyki

## Fizykochemiczna baza danych

naliza bezpieczeństwa systemów

# Nowa micha rynkowa

Sprzedaż i instalacja systemów komputerowych w polskich przedsiębiorstwach przemysłowych staje się w ostatnich latach ważnym źródłem dochodu wielu firm, nawet tych, które do niedawna wyłącznie składały komputery lub sprzedawały sprzęt i oprogramowanie. Dzięki poprawie kondycji niektórych sektorów polskiej gospodarki nie tylko dla nich, ale także dla integratorów, firm reklamowych, organizatorów targów oraz prasy pojawiła się nowa nisza rynkowa.

Po drugiej stronie lady znajdują się nabywcy, a więc przedsiębiorstwa przemysłowe różnej wielkości. Zatrudnieni w nich informatycy często po raz pierwszy stają się zaopatrzeniowcami i to od razu kupującymi za sumy sięgające setek tysięcy a nawet milionów dolarów. Zatem, będąc amatorami przystępują do negocjacji dobrze z wyszkolonymi handlowcami, silnie umotywowanymi procentem prowizji od ceny kontraktu.

W transakcjach zakupu systemu informatycznego przedsiębiorstwa koszty oprogramowania i sprzętu wynoszą przeciętnie 30% wartości kontraktu; wiedzy o tym, na co wydaje się pozostałe 70% środków nasi informatycy-zaopatrzeniowcy nie zdobędą przeglądając jedynie katalogi lub w trakcie rozmów z dostawcami.

Przed zakupem można skorzystać z pomocy firm doradczych. Nie są one tanie, ale próby przeprowadzania samodzielnych eksperymentów na wielką skalę mogą okazać się znacznie droższe. Oczywiście są przedsiębiorstwa, z własnym dobrym zespołem informatyków, którzy mogą znaleźć lepszy stosunek „wydajności do ceny" dla swojego systemu niż konsultanci zewnętrzni.

Specjaliści są zgodni, że najbardziej wartościowe dla podjęcia decyzji o wyborze własnej metody informatyzacji są wizyty w zakładach o podobnym profilu i wielkości. Mogą one zapobiec późniejszemu dostawianiu kolejnych komputerów i borykaniu się z problemami wielu sieci lokalnych. Niestety, firmy niezbyt chętnie zgadzają się na to, aby ktoś z zewnątrz poznał tak newralgiczny element ich działania jakim jest system zarządzania – nawet jeżeli nie są to pracownicy konkurencji.

Popularnym źródłem informacji (ale nie wiedzy) o systemach komputerowych dla przemysłu są oficjalne prezentacje produktów. Wadą takich pokazów jest pomijanie przez prelegentów milczeniem ograniczeń czy niedoskonałości swoich produktów, jak też zalet oferty konkurencji.

Są zespoły informatyków zakładowych, które dzięki swoim wystąpieniom na seminariach, publikacjom i zdobywanemu przez wiele lat doświadczeniu, mają w środowisku opinię ekspertów. Stworzyli oni grupę opartą na nieformalnych, koleżeńskich związkach. A oto przykład kilku porad, jakie w kameralnej, a nie w odświętnej targowej atmosferze, można uzyskać od praktyków z przemysłu:

- Mimo, że z firmą M pracujemy już kilkanaście lat, w ubiegłym roku nadużyli naszego zaufania i sprzedali nam ostatnią partię urządzeń produkowanych w starej technologii. Teraz ściągamy katalogi z centrali w USA i wiemy lepiej od ich przedstawicieli w Polsce co M ma do zaoferowania.
- Z firmą Y związaliśmy się jako ich pierwszy klient w Polsce. Mieliśmy trochę kłopotów ale za to niedrogo nas cały system kosztował. Teraz dostawca Z jest w podobnej sytuacji i dlatego możecie wynegocjować bardzo niską cenę, bo będziecie ich królikiem doświadczalnym, a w przypadku sukcesu – cennym punktem referencyjnym.
- W fabryce guzików pracuje z powodzeniem oprogramowanie firmy K, ale dowiedzcie się dlaczego po dwóch latach bolesnego wdrażania wycofano się ze współpracy z K w fabryce nici.
- Uważajcie na firmę B. Zapewniali nas, że uruchomią system F-K w trzy miesiące, a po dwóch latach nie widać końca (aktualnie temu odbiorcy – a jest to bardzo duże przedsiębiorstwo – grozi bankructwo z powodu błędów zarządzania).
- Negocjując z firmą D kontrakt na zakup oprogramowania, udało nam się uzgodnić cenę 10 razy niższą niż katalogowa.
- Konsultanci z K (300 USD/godzinę) wyszukiwali ciągle nowe problemy aby mieć u nas stałe zajęcie, a brak postępów był ich zdaniem wynikiem naszej nieudolności. Zrezygnowaliśmy z nich. Firmę K doradziła nasza centrala w USA, w której ten sam system działa z powodzeniem.
- Firma E ma dobry produkt i specjalistów, ale realizują zbyt wiele projektów równocześnie i dlatego nigdy nie mogą zmieścić się w uzgodnionych terminach.
- Firma doradcza X poleci wam dostawcę X1, który jest związany z X finansowo. O X1 piszą w samych superlatywach w prasie wydawcy X2, także zależnego finansowo od X. Wybierając od razu dostawcę X1 zaoszczędzicie na kosztach doradzania i na czasie zmarnowanym na czytanie nierzetelnych artykułów.
- Musieliśmy rozstać się z naszym wieloletnim dostawcą sprzętu i oprogramowania F, gdyż wykorzystując naszą zależność od swojej platformy stale podnosił ceny. Rozstanie było kosztowne, ale opłaciło się.
- Producent aplikacji A udostępnił nam jej kod źródłowy, więc możemy ją sami adaptować do naszych potrzeb. Uzgodniliśmy z A, że będziemy wdrażać tę aplikację w innych przedsiębiorstwach, także za granicą.
- Firma doradcza C potrafi napisać dwa różne opracowania na ten sam temat, ale oficjalnie pokazuje tylko jedno.

Czy znajdzie się organizator i sponsor, który zechce tą cenną, wysokim kosztem zdobytą wiedzę zagospodarować i upowszechnić?

Jerzy Dworecki

# informatyka

Miesięcznik
ISSN 0542-9951
INDEKS 36124


Ukazuje się elektroniczna wersja naszego miesięcznika. Oto nasz adres w Internecie: **www.pol.pl/informatyka/**


nr 6 1997
Czerwiec
Rok wydania XXXIII



REDAGUJE ZESPÓŁ:

dr Lesław WAWRZONEK
(redaktor naczelny)
mgr Jerzy SZYLLER
(redaktor)
mgr Tomasz SZEWCZYK
(redaktor)
mgr Zdzisław ŻURAKOWSKI
(redaktor)
mgr inż. Agnieszka ZIÓŁKOWSKA
(sekretarz redakcji)
Alina KLEPACZ
(redakcja techniczna, administracja)

KOLEGIUM REDAKCYJNE:

prof.dr hab. Leonard BOLC
mgr inż. Piotr FUGLEWICZ
prof. dr hab. Jan GOLIŃSKI
dr inż. Zenon KULPA
prof. dr inż. Jan MULAWKA
prof. dr hab. Wojciech OLEJNICZAK
mgr. inż. Jan RYŻKO
dr Witold STANISZKIS
dr inż. Jacek STOCHLAK
prof. dr hab. Maciej STOLARSKI
prof. dr hab. Zdzisław SZYJEWSKI
prof. dr hab. inż.
Ryszard TADEUSIEWICZ
prof. dr hab. Jan WĘGLARZ

PRZEWODNICZĄCY
RADY PROGRAMOWEJ
prof. dr hab. Juliusz Lech KULIKOWSKI



WYDAWCA:
Wydawnictwo Czasopism i Książek Technicznych SIGMA NOT Spółka z o.o.
ul. Ratuszowa 11
00-950 WARSZAWA
skrytka pocztowa 1004

REDAKCJA:
00-950 Warszawa,
ul. Ratuszowa 11, p. 628
skrytka pocztowa 1004
tel., fax: 619-11-61
tel.: 619-22-41 w. 159
e-mail: informat@pol.pl

PRENUMERATA:
tel. 40-35-86, 40-30-86



Materiałów nie zamówionych redakcja nie zwraca.

Autorzy artykułów proszeni są o przysyłanie tekstów na dyskietkach $3^1/_2$" - w czystych kodach ASCII (Latin II, Mazovia) lub edytorach: Word 2, Word 6.
Po szczegółowe informacje dla Autorów prosimy zwracać się do redakcji.

Redakcja nie ingeruje w treść i formę ogłoszeń i innych materiałów reklamowych, w związku z tym nie ponosi za nie odpowiedzialności.

**Ogłoszenia przyjmują:**
**- Redakcja, tel. 619-11-61**
**- Dział Reklamy i Marketingu**
**00-950 Warszawa, ul. Mazowiecka 12**
**tel.: 827-43-66, fax: 826-80-16**



Okładka:
AGAT, Jerzy Burski i Andrzej Jacyszyn,
Łamanie:
Alina Klepacz, program PageMaker 5.0
Druk:
Drukarnia SIGMA NOT Sp. z o.o.


# W numerze:


POLITECHNIKA ŚLĄSKA Biblioteka Główna w Gliwicach
P/1877/97



## INFORMACJE 2

## SYSTEMY INFORMACYJNE

**Kongres SAP** 6
**Polski sołtys w Ameryce** 6
Jerzy Szyller

## KONFERENCJE

**Techniki oddzielania grafiki od tekstu** 7
Beata Kwidzińska, Witold Malina

## PUBLIKACJE

**Architektury Klient-Serwer** 11
Krzysztof Jeżak

**Baza Danych Fizykochemicznych i Termofizycznych** 19
Marcin Gorawski, Jacek Frączek

### Systemy czasu rzeczywistego

**Wprowadzenie do cyklu** 25
Janusz Górski

**Analiza drzew błędów** 28
Andrzej Wardziński, Marko Čepin

## OPINIE

**Fatamorgana, mechanizmy napędowe i korzenie informatyki** 33
Zygmunt Ryznar

## BIULETYN PTI 39

## Targi CeBIT'97 - kilka refleksji

**CeBIT** to coroczne spotkanie producentów i przedstawicieli instytucji naukowych z branży informatycznej, elektronicznej i telekomunikacyjnej. Podstawowe cele Targów to prezentacja najnowszych osiągnięć w danej branży w okresie minionego roku oraz promocja wybranych produktów. Są to największe targi komputerowe na świecie (6855 wystawców z 59 krajów i ponad 600 000 zwiedzających z całego świata w ciągu 7 dni w tym roku), na których firmy, instytucje naukowo-badawcze i organizacje zajmujące się przepływem technologii i gospodarką nie tylko prezentują swoje osiągnięcia i zamierzenia, ale również nawiązują kontakty, zawierają kontrakty i promują swoje produkty.

Ze względu na olbrzymie zainteresowanie Targami i związanymi z nim kłopotami organizacyjnymi, w 1995 roku zostały one rozdzielone na „CeBIT Professional", odbywające się w marcu, oraz „CeBIT Home" odbywające się co dwa lata jesienią (pierwszy raz odbyły się w 1996 r.). Targi profesjonalne są podzielone tematycznie zwykle następująco:

- Technika informacyjna (np. sprzęt komputerowy i oprogramowanie systemowe, narzędziowe i użytkowe);
- Sieci komputerowe;
- Wytwarzanie wspomagane komputerowo (CIM);
- Oprogramowanie, doradztwo i usługi (informatyczne, rozumiane bardzo szeroko, np. promocja środków do czyszczenia komputerów);
- Telekomunikacja (dział bardzo w ostatnich latach ekspansywny);
- Technika biurowa;
- Banki i kasy oszczędnościowe (w tym rozwiązania związane z bezpieczeństwem systemów informatycznych);
- Badania i nauka.

Od przedstawionego powyżej podziału tematycznego (w zależności od prezentowanej oferty firm), mogą zaistnieć w poszczególnych latach niewielkie odstępstwa.

Chociaż są to targi międzynarodowe, to ze względu na ich lokalizację najwięcej wystawców i zwiedzających jest z Niemiec. Również wielkie, ponadnarodowe firmy prezentują się zwykle poprzez swoje niemieckie przedstawicielstwa.

Firmy polskie prezentowały się w ramach wspólnej ekspozycji firm polskich organizowanej przez Polską Izbę Informatyki i Telekomunikacji za pośrednictwem Centrum Promocji Informatyki. Ekspozycja ta jest od trzech lat umiejscowiona w hali nr 5, na tym samym obszarze i w podobnym wystroju, co sprawia, że można mówić o pewnej ciągłości ekspozycji polskiej na Targach CeBIT.

Polskie firmy prezentujące swoje produkty w tym roku na CeBIT-cie to:

**Arkus Electronics, Axxel Computer Sp. z o.o., Comex Joint Venture Ltd., Digilab Sp. z o.o., Elzab S.A., Fideltronik, Zakład Elektroniki Górniczej S.A., Contec Telecom Control Technology, DGT Sp. z o.o., Satel S.A., Vector Sp. z o.o.**

Na targach była prezentowana także nauka polska, ponieważ jedno ze stanowisk wystawowych w hali 5 współdzieliły **Wojskowa Akademia Techniczna i Instytut Badań Systemowych z Warszawy oraz Akademia Ekonomiczna z Poznania**. Wraz z Zakładem Elektroniki Górniczej stoisko współdzielił **Instytut Informatyki Politechniki Śląskiej**. Koszty 1m² powierzchni wystawowej w 1997 roku to 810 DM (jeszcze w styczniu br. koszty wynajmu powierzchni wystawowej zostały w ok. 50% zrefundowane przez Ministerstwo Współpracy z Zagranicą). Koszty zabudowy 1m² wyniosły 150 DM. W sumie ekspozycja polska w hali 5 zajęła powierzchnię 120m².

Czy polska ekspozycja była duża, jak na CeBIT-owe miary? Dla porównania *Siemens Nixdorf* , w tym roku zajął 5000m² powierzchni wystawowej, biurowej i warsztatowej. Pracowało na niej 1200 pracowników firmy i 130 hostess oraz 300 dodatkowych pracowników firmy mobilizowanych na szczególnie „gorące" chwile. Na czas wystawy przygotowano 100 000 filiżanek kawy, 14 000 kiełbasek oraz 1,5 tony owoców dla pracowników i gości. Ale w końcu Siemens to sztandarowy niemiecki koncern zajmujący się „także" informatyką.

Skoro jesteśmy przy liczbach: w tym roku swoje sukcesy komercyjne święcił Taiwan reprezentowany przez 401 firm, co uplasowało go na drugim (po USA - 446) miejscu wśród zagranicznych wystawców.

W roku 1997 (podobnie jak i w trzech poprzednich latach) motorem napędzającym koniunkturę na rynku produktów informatycznych była telekomunikacja. Szczególnie jaskrawym tego przykładem jest walka o klienta toczona z niebywałym rozmachem (również na CeBIT-cie) przez firmy zajmujące się telefonią komórkową. Hasło „Mobile Communication" można było zauważyć na wielu stoiskach wystawowych, nie tylko w halach 14 (tutaj prezentowały się cztery firmy polskie), ale także w tych, które tradycyjnie są przeznaczone dla wystawców prezentujących wyroby związane z szeroko pojętą telekomunikacją. Szczególnie dużo mówiło się o *Deutsche Telekom* w kontekście zawartej umowy o współpracy pomiędzy *DT* i *Microsoft Corp.* w zakresie wspólnego promowania usług T-Online i Microsoft Network (MSN). Zresztą walka o klienta poprzez wyszukane formy promocji, stawki opłat telefonicznych oraz wyposażanie coraz mniejszych telefonów przenośnych w coraz wymyślniejsze gadżety nikogo już tutaj nie dziwi.

Z branży telekomunikacyjnej wart jest odnotowania również pokazywany po raz pierwszy w Europie system wideo konferencyjny LivePhone prezentowany przez firmę *PictureTel*. System ten wykorzystuje analogowe łącza telefoniczne (dotychczas wideo konferencje wymagały łączy ISDN).

W hali nr 7 można się zapoznać z najnowszymi produktami i trendami w produkcji drukarek. W hali nr 8 z kolei, zgromadzeni są wszyscy którzy zajmują się produkcją pamięci zewnętrznych. Tutaj jest wart wzmianki produkt firmy *Seagate* „Elite 23" - dysk o pojemności 23GB (10 talerzy dyskowych, 10 000 obr/min).

Tradycyjnie, nauka (uniwersytety i instytuty naukowo-badawcze) prezentują się w hali 22. Tutaj można zapoznać się z projektami badawczymi i ich wynikami realizowanymi w uczelniach i instytutach naukowo-badawczych głównie z terenu Niemiec, chociaż można też znaleźć uczelnie z krajów niemieckojęzycznych, a nawet z Ukrainy. Ekspozycje uczelni niemieckich były pogrupowane według regionów (niem. Land). Tutaj znalazło się też stanowisko Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie (Ministerstwo Szkolnictwa, Nauki, Badań i Technologii) na którym oferowano raporty o zrealizowanych projektach naukowo-badawczych i planowanych przedsięwzięciach w zakresie nauki i rozwoju.

W hali 23 są firmy których głównym polem działalności jest bezpieczeństwo informacji. Tutaj można zapoznać się z najnowszymi osiągnięciami w dziedzinie oprogramowania antywirusowego, metod fizycznej kontroli dostępu osób do informacji (wiele firm prezentowało rozwiązania wykorzystujące karty magnetyczne do kontroli dostępu), metod zabezpieczania informacji przed awariami sprzętu (backup i macierze RAID) itd. Niewątpliwym „kąskiem" dla znawców jest stanowisko wystawowe Bundesamt für Sichercheit in der Informationstechnik (urząd państwowy ds. bezpieczeństwa w technikach informacyjnych) na którym jest prezentowany „państwowy" sposób widzenia szeroko rozumianego bezpieczeństwa informacji (od podsłuchów poprzez awarie sprzętu do unormowań

w zakresie nadawania państwowych certyfikatów bezpieczeństwa).

Ten rok nie przyniósł specjalnych rewelacji (trudno bowiem za taką uznać architekturę MMX mocno na targach eksponowaną, ale mającą swoją premierę jeszcze przed CeBIT-em), jaką np. w zeszłym roku była dyskietka o pojemności 120MB i napęd dla niej, na którym można także zapisywać i odczytywać „klasyczne" dyskietki 3,5". Tak na marginesie, rozwiązanie to umacnia swoją pozycję, ponieważ do firmy Compaq dołączył Siemens Nixdorf oraz Fujitsu-ICL. Również Microsoft, Award i AMI zaczęli wspierać swoimi produktami LS-120.

Na tegorocznym CeBIT-cie obficie było prezentowane oprogramowanie typu GIS. Początki tego boomu (dla Niemiec prognozy mówią o 30% wzroście na oprogramowanie GIS w tym roku) można było zauważyć już na CeBIT'94. Obecne aplikacje są już dojrzałe, choć dla wielu z nich zasięg map kończy się na Odrze, a w najlepszym razie na Bugu.

Kolejny popularny tegoroczny temat to Internet. Chciałoby się śtrawestować znany slogan „Internet to komputer", tym bardziej że nawet tak mocno software'owa firma jak Corel zaprezentowała swój pierwszy komputer - VNC - Video Network Computer, będący nawiązaniem do objawionej w zeszłym roku przez firmę Sun, koncepcji komputera sieciowego. W Niemczech germany.net wzięła chyba przykład z TP S.A. i oferuje bezpłatny (tzn. płacimy tylko za połączenia telefoniczne) dostęp do Internetu. Diabeł jak zwykle tkwi w szczegółach - nawigacja po tym bezpłatnym Internecie jest przerywana (jak w prywatnej telewizji) reklamami.

Potwierdzają się prognozy odnośnie CeBIT Home - „wielcy" branży informatyczno-elektronicznej, produkujący również sprzęt typu „domowo-rozrywkowego" są zmuszeni wystawiać się na obu CeBIT-ach. Panasonic wystawił oferowany na rynku niemieckim za ok. 1400DM DVD-Player (podobnie jak Sony i Toshiba, ale ich produkty są dostępne na razie tylko na rynku japońskim). W tym roku szczególnie widoczne były ekspozycje naszych sąsiadów zza wschodniej granicy. Urządzone z rozmachem stoiska Rosji i Ukrainy (rozmieszczone w kilku halach) prezentowały interesujące produkty i przyciągały tłumy zwiedzających.

W tym roku Deutsche Messe AG w Hanowerze obchodzą swoje 50-cio lecie. Ta, pomimo swojego wieku wciąż dynamiczna instytucja, przygotowuje się do wystawy EXPO 2000, która w 2000 roku będzie się odbywała na hanowerskich terenach wystawowych.

Jednak również i tak doświadczona instytucja jak Deutsche Messe AG nie ustrzegła się wpadki. Na CD-ROM-ie „What's new at CeBIT" wyprodukowanym przez Hardlight Multimedia Ltd (sponsorowanym przez Macromedia) i rozprowadzanym z „Gazetą Targową", znalazł się wirus makr Worda „Wazzu"!

Kończąc refleksje na temat CeBIT-u wszystkim zainteresowanym prezentacją własnego dorobku na Targach chciałbym przypomnieć, że wraz z zakończeniem tegorocznych targów rozpoczęły się przygotowania do Targów CeBIT'98, które odbędą się w dn. 19-25.03.1998. Oczywiście w Hannowerze!

**Krzysztof Liderman**

## II Międzynarodowa Konferencja BIS '98

II Konferencja Business Information Systems (BIS) odbędzie się w Poznaniu podczas Międzynarodowych Targów INFOSYSTEM'98 w dniach 22 -24 kwietnia 1998 r. Jej przewodniczącym jest prof. Witold Abramowicz. Patronat prasowy nad konferencją sprawuje m.in. „Informatyka". Celem BIS jest stworzenie warunków do dyskusji na temat budowy, wdrażania, stosowania i doskonalenia systemów informatycznych dla procesów gospodarczych. Tematyka konferencji obejmuje m.in.: dziedzinowe systemy informacyjne, integrację systemów informacyjnych oraz transakcje elektroniczne. Termin przesłania artykułów do Komitetu Programowego mija 10 listopada 1997 r.

Informacje na temat konferencji **BIS** udostępniają: **Komitet Organizacyjny** przy Akademii Ekonomicznej w Poznaniu, **tel. (061) 869 92 61, fax (061) 866 89 24, e-mail bis@sylaba.poznan.pl, URL: http://www.bis.pozn.com.**

Komitet Organizacyjny przy MPT: Biuro Informacji Technicznej – MPT, ul. Głogowska 14, 60-734 Poznań, tel. (061) 869 22 06, fax (061) 866 66 50.

## III Międzynarodowa Konferencja INTEGRACJA CZŁOWIEK-KOMPUTER (HCI'97)

Konferencja HCI'97 organizowana przez Katedrę Informatyki Ekonomicznej Uniwersytetu Gdańskiego pod auspicjami Ministerstwa Edukacji Narodowej oraz SAP Polska odbędzie się w Sopocie 16-18 października 1997 r.

Celem konferencji jest stworzenie możliwości prezentacji metod, technik i narzędzi oraz ich zastosowania (re-engineering, outsourcing, wirtualizacja) w rekonstrukcji i modelowaniu procesów gospodarczych, a także wykorzystanie zintegrowanych systemów informacyjnych klasy MRP II w tych procesach.

Teksty referatów i zgłoszenia udziału w Konferencji należy nadsyłać do 30 czerwca br. na adres:

hci@panda.bg.univ.gda.pl. Informacje o formie referatu i prezentacji są dostępne pod adresem:

http://www.bg.univ.gda.pl./hci.

## Informatyka dla lekarzy

Druga konferencja „Techniki informatyczne w medycynie" odbędzie się w dniach 16-18 października w Ustroniu-Jaszowcu. Celem konferencji, dofinansowanej przez KBN, jest umożliwienie wymiany doświadczeń specjalistom nauk medycznych i technicznych w zakresie informatyki medycznej. Głównymi tematami konferencji TIM'97 będą: biopomiary, biotechnologie, systemy inteligentne, multimedia i bazy danych. Przewodniczącym Komitetu Organizacyjnego konferencji jest prof. Jan Piecha z Uniwersytetu Śląskiego. „Informatyka" objęła patronat prasowy nad TIM'97. Zainteresowani udziałem w konferencji proszeni są o skontaktowanie się z jej sekretariatem: tel.:(0-32) 1918087, e-mail: piecha@us.edu.pl

## BankForum 97

3-4 czerwca br. odbyła się Konferencja BankForum 97 na temat roli informatyki w bankowości. Uczestnikami konferencji byli szefowie większych polskich banków, dyrektorzy departamentów informatyki w bankach oraz przedstawiciele czołowych instytucji finansowych i konsultingowych. Na konferencji dyskutowano nt. kierunku dalszego rozwoju informatyki bankowej oraz trendów ewolucji bankowości wynikających z dynamicznego rozwoju internetu oraz intranetu.

## AXIS StorPoint CD/T Dual SCSI

Serwer CD-ROM Axisa z dwoma portami SCSI obsługuje do 14 bezpośrednio podłączonych napędów CD-ROM lub do 112 napędów CD-ROM podłączonych przy użyciu SCSI expanderów LUN (Logical Unit Number). Cały system może być wykorzystany przez wiele aplikacji,

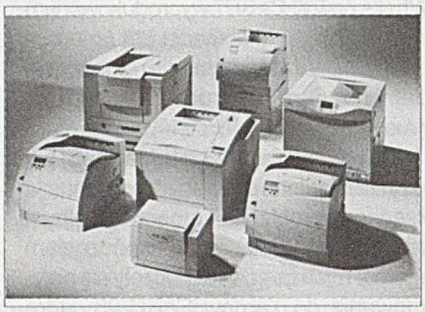
np. do wykorzystania danych lub oprogramowania przez grupy robocze dowolnych rozmiarów. Serwer podłączany do sieci Ethernet pracuje niezależnie od serwerów plików. Serwer ten umożliwia jednoczesną obsługę wszystkich sieciowych protokołów i systemów operacyjnych (NetWare, Windows NT/95/3.xx, Unix i OS/2, jak również HTTP). W rezultacie użytkownicy mają równoczesny dostęp do informacji zapisanych na płytach CD-ROM, niezależnie od systemu operacyjnego z jakiego korzystają. Zintegrowany Web Server pozwala udostępnić w sieci intranet lub internet informacje przechowywane na płytach CD-ROM w formacie HTML. Serwer ten może być konfigurowany i zarządzany za pomocą dowolnej przeglądarki WWW, niezależnie od platformy sieciowej. Przepustowość nowego serwera CD-ROM sięga 1,100 kB/s.

## Specyfikacja Speaker Veryfication API (SVAPI)

Speech Recognition API Committee poinformowała o opracowaniu technologii umożliwiającej dodatkowe zabezpieczenie komputerów i aplikacji poprzez identyfikowanie i weryfikowanie głosu użytkowników - SVAPI. Specyfikacja API pomaga w tworzeniu oprogramowania potrafiącego zidentyfikować głos i skontrolować prawa dostępu jego właściciela do informacji i innych zasobów komputerowych. Można ją wykorzystać w aplikacjach z takich dziedzin, jak telekomunikacja, handel elektroniczny, bankowość i finanse. Otwarta standardowa biblioteka API pozwoli na projektowanie aplikacji zgodnych z systemami różnych producentów (włączając istniejące systemy sieciowe i internetowe) i mogących z tymi systemami współpracować. Specyfikacja SVAPI jest bezpłatna i można ją skopiować ze strony WWW: http://www.srapi.com/svapi.

## Nowe drukarki Lexmarka

Firma Lexmark zaprezentowała w Warszawie nowe typy swoich drukarek. Rodzina monochromatycznych drukarek laserowych Optra S składa się z 3 drukarek o różnej wydajności (12, 16 i 24 str/min) i Optra SC wyposażona w adapter MarkNet S do pracy w sieci. Wszystkie drukują z rozdzielczością 1200x1200 dpi. Nowe drukarki charakteryzuje modularność wyposażenia dodatkowego, a w tym łatwego podłączania kilku podajników z różnego typu papierem. Pokazano też nową, tanią



drukarkę atramentową do zastosowań domowych i biurowych 3000 Color Jetprinter w cenie ok. 9 mln zł (bez VAT), a także niedrogą drukarkę laserową o rozdzielczości 600x600 dpi i miesięcznej wydajności 10 tys. stron, w cenie ok. 1500 zł (bez VAT). Jak twierdzi Robert Milczyński z przedstawicielstwa Lexmarka w Polsce, największą zaletą drukarek jego firmy są, oprócz dobrej jakości, niskie w porównaniu z konkurencją koszty eksploatacji. (lw)

## Komputer dla niewidomych

Siemens Nixdorf współpracując z laboratorium badawczo-rozwojowym C-LAB z Padernborn opracował system oparty na komputerze osobistym wyposażonym w wyświetlacz pisma Braille'a oraz syntezator mowy. Informacje sterujące graficznym interfejsem użytkownika są zamienione w informacje tekstowe i tłumaczone na zapis Braille'a. Dane wprowadzane są za pomocą głosu lub klawiatury. Dzięki programom adaptacyjnym niewidomi mogą używać systemów graficznych, takich jak Windows 3.x, 95, NT lub OS/2. Wyposażenie komputera w modem oraz kartę ISDN lub sieciową umożliwia dostęp do internetu.

## Intergraph dla floty amerykańskiej

Avondale Alliance projektuje statek bojowy LPD-17 dla U.S. Navy. Intergraph Corporation jest partnerem Avondale Alliance i dostarcza całą infrastrukturę informatyczną do tego projektu. Firma będzie implementować system w postaci IPDE (Integrated Product Data Environment), który zawiera oprogramowanie do wspomagania projektowania (CAD), wspomagania produkcji (CAM), wspomagania inżynierii (CAE), zarządzania dokumentacją i inne elementy niezbędne w realizacji projektu.

## Nowy adapter sieciowy - EtherPower II 10/100

Najnowszym osiągnięciem firmy **SMC** w dziedzinie adapterów sieciowych jest EtherPower II 10/100. W karcie zastosowano nowoczesną technologię EPIC polegającą m.in. na implementacji do układów scalonych funkcji jednoczesnej transmisji w obie strony (full-duplex), ustalenia kolejek transmisji (transmit chaining), współpracy z 32-bitową szyną główną.

**Karta EtherPower II 10/100 sama rozpoznaje tryb działania sieci** i automatycznie dostosowuje się do niego, również po dokonaniu *upgrade'u* sieci z 10 do 100 Mbps. Sama też wybiera najszybszy dostępny kanał transmisyjny i optymalny tryb działania. Karta konfiguruje się sama w systemie Windows 95 lub Windows NT z pomocą SuperDisk. Pracuje z systemami operacyjnymi: Novell NetWare 3.x/4.x, DOS, OS/2, Windows 95/ NT/3.x, Windows for Work- groups, Banyan Vines, SCO Unix, IBM LAN Server, MacOS System7 (PowerMAC), AIX PowerPC i in. Opcjonalnie można zainstalować Flash ROM z kontrolą wirusową.

## Czytnik CD-ROM o 18-krotnej prędkości

AcerMagic 616A firmy Acer Components ma bezpośredni dostęp do pamięci (DMA - Direct Memory Access), co daje możliwość przekazania informacji bezpośrednio z dysku CD do pamięci, z pominięciem procesora. W czytniku zastosowano technologie stałej prędkości kątowej (CAV - Constant Angular Velocity) i liniowej (CLV - Constant Linear Velocity), które utrzymują parametry odtwarzania bez względu na to, która ścieżka jest aktualnie odczytywana. Czytnik mieści się w standardowej obudowie do napędów 5,25 cala, współpracuje z interfejsem EIDE/ATAPI i jest zgodny z techniką „plug-and-play" Widows 95/NT.

## Composter na polskim rynku

Hector i Texas Instruments Software/ Sterling Software podpisali porozumienie, które daje firmie Hector wyłączność na sprzedaż zaawansowanego oprogramowania narzędziowego Composer na terenie Polski. Kompleksowe systemy informatyczne zrealizowane przy pomo-

cy Composer'a pracują w sektorach: bankowości, ubezpieczeń, telekomunikacji i wojskowym. Pakiet umożliwia tworzenie aplikacji klient-serwer różnego typu. Composer działa i generuje oprogramowanie, które może być uruchamiane w różnych środowiskach sieciowych, systemowych i sprzętowych.

## Specjalna oferta akademicka

Firma **StatSoft** proponuje szkołom wyższym licencje typu Site License, co ma na celu udostępnienie studentom oraz pracownikom naukowym i dydaktycznym legalnych kopii pakietów STATISTICA i SIS (STATISTICA INDUSTRIAL SYSTEM). Opłata roczna (forma dzierżawy licencji) przy liczbie użytkowników w przedziale 25 - 50 wynosi 100 USD, a opłata jednorazowa (zakup licencji) za 5 użytkowników wynosi 3125 USD.

## Otwarte standardy drukowania w internecie

**Novell** wraz z innymi firmami opracowuje protokół Internet Print Protocol (IPP) dla operacji drukowania w intranetach i internecie. Protokół uprości drukowanie w niejednorodnych środowiskach sieciowych oraz zapewni standaryzację drukowania zdalnego. Użytkownicy będą mogli drukować dokumenty bezpośrednio na drukarkach znajdujących się w dowolnym miejscu w internecie, wykorzystując połączenia internetowe.

## Zabezpieczenia sieci korporacyjnej

Clico Centrum Oprogramowania uzyskało prawo do sprzedaży produktów firmy SunSoft (firma wchodząca w skład Sun Microsystems) wyposażonych w algorytmy szyfrujące, objęte dotychczas restrykcjami eksportowymi Rządu USA. Prawo do nabycia tych produktów uzyskuje się na drodze indywidualnego zezwolenia przydzielanego dla określonego użytkownika. Oferta dotyczy produktów wchodzących w skład rodziny Solstice Security Software:

- Solstice Firewall-1 Version 3.0 DES Encryption Module
- Solstice SunScreen EFS 1.0
- Solstice SunScreen SKIP Encryption Client.

## X Ogólnopolskie Forum Teleinformatyki

W Starych Jabłonkach na Mazurach odbyło się w końcu maja jubileuszowe X OFTI. Jest to coroczne spotkanie szefów instytucji, organizacji i firm teleinformatycznych, organizowane obecnie przez Polską Izbę Informatyki i Telekomunikacji (PIIiT) oraz Centrum Promocji Informatyki. Przed każdym Forum określane są inne grupy poruszanych tematów, które łączy to, że mają istotne znaczenie dla rozwoju teleinformatyki w Polsce. W tym roku pierwszy dzień poświęcono teleinformatyce i społecznym, prawnym oraz gospodarczym warunkom jej rozwoju. Problemy związane z tą tematyką prezentowali przedstawiciele PIIiT - prezes Wacław Iszkowski, wiceprezes Aleksander Szwarc i członek prezydium Tomasz Chlebowski oraz przedstawiciele Biura Bezpieczeństwa Narodowego – Wiesław Paluszyński i Marek Dukaczewski.

W drugim dniu mówiono o pieniądzach (choć byli sami dżentelmeni), a więc o warunkach finansowania przedsięwzięć teleinformatycznych, w tym przede wszystkim o możliwościach i pożytkach wejścia firmy na giełdę. Mówili o tym przedstawiciele ComputerLandu, Apeximu, ATM, Softbanku, Macrosoftu i Optimusa. W drugiej sesji tematycznej tego dnia, skupiono się na technikach promocji firm teleinformatycznych przez media.

Ostatni dzień poświecono relacjom polityki i teleinformatyki. Jak się okazało dyskusja prowadzona przez z-cę red. nacz. Rzeczpospolitej Bożenę Wawrzewską ograniczyła się do haseł wyborczych i sposobu wykorzystywania komputerów PC i notebook'ów przez zaproszonych polityków. O swoich związkach z komputerami mówili Waldemar Pawlak, Janusz Korwin-Mikke i Krzysztof Król.

Niewątpliwie wydarzeniem Forum i ucztą intelektualną było wystąpienie prof. Władysława M. Turskiego – matematyka, pierwszego prezesa Polskiego Towarzystwa Informatycznego. Profesor, jak to ma w zwyczaju, przedstawił kontrowersyjne spojrzenie na wiele spraw, które do czasu spotkania z nim wydawały się wielu ludziom oczywistymi. Profesor mówił o tym, że rozwijająca się w szalonym tempie teleinformatyka, aby mogła na siebie zarobić w wyniku obniżki kosztów własnych, musi wywołać w społeczeństwach nowe potrzeby. Drogą kolejnych eliminacji prof. Turski doszedł do wniosku, że taki dochód zapewni wyłącznie dostarczanie rozrywki. Drugą ważną tezą wystąpienia Profesora była potrzeba ingerencji państwa w rozwój globalnych sieci, a zwłaszcza w to, co po niej wędruje. Prof. Turski mówił, że niecenzurowanie tradycyjnych listów pocztowych nie może się prosto skalować na wolność słowa w informacjach przesyłanych w internecie, bo w tym drugim przypadku „siła rażenia" jest nieporównywalnie większa. Użył tu barwnego porównania tzw. korkowca do haubicy - oba te urządzenie robią podobne „bum", a niekiedy korkowiec przyłożony do oka może uczynić nawet większe szkody, to jednak posiadanie haubicy wymaga zezwolenia.

Z prawej prof. Władysław M. Turski

*X OFTI było jak zwykle imprezą ważną i udaną, choć są oczywiście sceptycy, którzy twierdzą, że nie całkiem. Szkoda, że w części „Polityka i informatyka" nie próbowano skłonić zaproszonych znanych polityków do wypowiedzenia się na tematy systemów informatycznych w skali Państwa. Nie mówiono o krajowych systemach informatycznych, a więc o milionach dolarów na nie wydawanych (czasem marnowanych) jak też o miliardach dolarów traconychprzez budżet (Polaków) dlatego, że tych systemów nie ma.*

*Poglądy wygłoszone przez prof. Turskiego dały słuchaczom wiele do myślenia, a jego dyskusja z politykami pokazała, że trudno czasem odróżnić polityka od doktrynera.*

*Zabrakło na X OFTI zagadnień związanych z teleinformatyką w administracji, ale z pewnością temat ten powróci w przyszłym roku.*

**Lesław Wawrzonek**

# Kongres SAP

**Rozwiązania dla przedsiębiorstw** to nazwa I Ogólnopolskiego Kongresu SAP, który odbył się w Warszawie w maju 1997. SAP jest jedną z bardziej liczących się firm, na światowym a także na polskim rynku w zakresie produkcji oprogramowania wspomagającego zarządzanie przedsiębiorstwem i metodologii jego wdrażania.

Spotkanie z przedstawicielami firmy SAP, jej partnerami i klientami miało formę sesji plenarnych i wykładów prowadzonych przez specjalistów, konsultantów, oraz praktyków zajmujących się problematyką informatyzacji przedsiębiorstw i organizacji gospodarczych.

## SAP R/3 nie tylko dla bogatych

Dr Alfred Wenzel, dyrektor generalny SAP Polska otworzył Kongres SAP'97, który ma się stać corocznym forum wymiany myśli ekonomicznej i informatycznej w naszym kraju. Na wstępie dr Wenzel starał się obalić kilka obiegowych opinii na temat oprogramowania SAP R/3. Twierdzi się np., że oprogramowanie SAP jest zbyt drogie dla przeciętnej polskiej firmy. Mówca zapewnił, że każdą firmę wdrażającą rozważnie i stopniowo system informatyczny jest stać na zakup modułów systemu SAP R/3.

## Jakoś to będzie!

Kolejną sesję prowadzili dyrektor Deloitte&Touche Lesław Paga oraz prof. SGH Janusz Beksiak. L. Paga kreślił pesymistyczny obraz stanu polskiej gospodarki. Zagrażają jej – jego zdaniem – monopole w sektorze paliwowym, telekomunikacji i przemyśle wydobywczym oraz brak prób restrukturyzacji rolnictwa. Struktury te w obecnej postaci nie mają szans w konkurencji z przedsiębiorstwami Unii Europejskiej po przystąpieniu do niej Polski. Lesław Paga jedyną nadzieję upatruje w tak często przez Polaków przywoływanym powiedzeniu, że „jakoś to będzie".

Profesor Janusz Beksiak – wróg etatyzmu – dowodził, że zamiast zmniejszać się, etatyzm w dzisiejszej Polsce rośnie, jak w minionych czasach realnego socjalizmu. Profesor jest zdania, że jeśli ta tendencja nie zostanie odwrócona, to rozwój gospodarczy zostanie po prostu zdławiony.

## Konsulting przede wszystkim

Kongresowi towarzyszyła ekspozycja partnerów SAP wspomagających na różnych etapach wdrażania systemów informatycznych. Większość z nich to firmy doradcze, takie jak Andersen Consulting, APCON, Coopers&Lybrand, DGA, IBM, IMG, INIT, KPMG, NOVA czy Price Waterhouse. Atrakcją nielicznych firm producentów na ekspozycji był Sun Microsystems z sieciowym komputerem (Network Computer - NC), na którym działało oprogramowanie SAP R/3 w środowisku Javy.

## Było panelistów wielu...

W sesji plenarnej pod hasłem „Strategia dla polskich przedsiębiorstw" udział wzięli: prof. Witold Abramowicz z AE w Poznaniu, Władysław Baka z Bankowego Funduszu Gwarancyjnego oraz prezes PSL Waldemar Pawlak i były minister dr Janusz Lewandowski z IBGR. Przebieg sesji moderował Andrzej K. Wróblewski z Polityki.

Dr J. Lewandowski stwierdził, że na zahamowanie dynamiki polskiej gospodarki w ostatnim okresie wyraźnie wpływa ograniczenie procesu prywatyzacji. Prezes Waldemar Pawlak przedstawił zalety internetu i narzędzi informatyki dla prowadzenia gospodarki i polityki Państwa. Jego wystąpienie dowiodło, że mimo ostrożności przejawianej w kontaktach gospodarczych ze światem zachodnim, jest zwolennikiem integracji informacyjnej i wykorzystania internetu do prowadzenia polityki.

## Najpierw biznes, potem informatyka

Wasa Rahmana z polskiego oddziału Andersen Consulting przedstawił wykład o reorganizacji procesów biznesowych (BPR). Wykład drugi, prowadzony przez wicedyrektora Coopers&Lybrand, Mieczysława Grynia dotyczył kwestii zasadniczej - dlaczego BPR jest najważniejszy dla przedsiębiorstw i organizacji gospodarczych. Stwierdził on, że osiągnięcie i utrzymanie przewagi konkurencyjnej można osiągnąć jedynie w drodze innowacyjnego przeprojektowania i wdrożenia zmian w podstawowych procesach biznesowych. W tym kontekście informatyzacja, aczkolwiek konieczna, schodzi na drugi plan, po przeprowadzanej pod kątem biznesowym reorganizacji przedsiębiorstwa. Uważam, że jest to najważniejsza myśl wypowiedziana w ciągu obu kongresowych dni.

*Jerzy Szyller*

# Polski sołtys w Ameryce

Frank Soltis (vel Sołtys), główny architekt najlepiej sprzedającego się na świecie w ostatnim dziesięcioleciu systemu firmy IBM AS/400 gościł w Warszawie. Dr Soltis zrobił w Ameryce wspaniałą karierę. Zatrudniony przez wiele lat w laboratoriach badawczych firmy IBM stał się autorytetem w dziedzinie architektury systemów komputerowych i organizacji mikroprocesorów. W ostatnich latach współuczestniczył w definiowaniu architektury procesora PowerPC, który ostatecznie stał się sercem 64-bitowego systemu PowerPC AS RISC.

Podczas seminarium Frank Soltis skupił się na pokazaniu podstawowych kierunków rozwoju współczesnych mikroprocesorów i architektur komputerowych. Główną tezą, która ciągle przewijała się w prezentacji były lawinowo rosnące koszty opracowania i produkcji dzisiejszych mikroprocesorów. Dopóki mikroprocesor zawierał kilkadziesiąt lub co najwyżej kilkaset tysięcy tranzystorów w układzie nakłady na badania, rozwój i budowę nowych linii produkcyjnych nie przekraczały kilku setek milionów dolarów. Koszt takiego przedsięwzięcia dzisiaj, gdy układ zawiera miliony tranzystorów, a stosowana litografia masek zeszła do poziomu poniżej 0,5 mikrona sięga 10 miliardów dolarów. Nawet giganci komputerowi tacy jak IBM nie są w stanie wytrzymać sami takich wydatków jednocząc się z innymi firmami, jak w przypadku procesora PowerPC z Motorolą i Apple.

Mimo to dr Soltis nadal nie widzi przeszkód, aby prawo Moore'a, twórcy pierwszego mikroprocesora, stanowiące, że co 18 miesięcy liczba tranzystorów w mikroprocesorach ulega podwojeniu, miało stracić ważność. Proces wzrostu liczby podstawowych elementów procesora będzie trwał nadal. Naczelny architekt AS/400 jest przekonany, że za parę lat układy zawierające 100 mln. tranzystorów, a więc o wydajności 20 procesorów Pentium, będą czymś oczywistym. Dalszy rozwój technologii powinien doprowadzić do liczby miliarda elementarnych układów w ramach jednego „chipu". Natomiast lawinowy wzrost kosztów badań i produkcji takich układów doprowadzi do zaniechania przez wiele firm ich wytwarzania. Dr Soltis przewiduje, że na początku XXI wieku na placu boju pozostanie tylko Intel z Pentium i IBM z PowerPC.

Frank Soltis zapytany jakie architektury będą odpowiadały przyszłym wyzwaniom odpowiedział, że będą to niewątpliwie struktury wieloprocesorowe oparte o procesory z własnymi pamięciami lub działające na wspólnej pamięci. Przytoczył też przykłady obu rozwiązań realizowane ostatnio na zamówienie rządu amerykańskiego. Pierwszy z systemów o wydajności jednego Teraflopa ($10^{12}$ operacji zmiennoprzecinkowych na sekundę) zaprojektowała firma Intel, natomiast drugi firmy IBM o wydajności trzech Teraflopów będzie dzięki specjalnej superprzełącznicy łączył tysiące procesorów ze wspólną pamięcią operacyjną. Mówiąc o tych niezwykłych komputerach nasz rodak zasugerował, że granice wydajności współczesnych komputerów są niemożliwe obecnie do określenia.

**Jerzy Szyller**

# Techniki oddzielania GRAFIKI od tekstu

Beata Kwidzińska, Witold Malina

Przedstawiono przegląd metod oddzielania grafiki od tekstu oraz ograniczenia tych metod. Omówiono podstawowe etapy analizy dokumentów typowe dla większości stosowanych obecnie metod.

W ostatnich latach obserwujemy coraz więcej zastosowań systemów komputerowych do przetwarzania dokumentów (przez dokument będziemy rozumieć każdą stronę zawierającą tekst, rysunki, zdjęcia itp.). Wraz z rozwojem tego typu systemów pojawił się problem co zrobić z istniejącą już dokumentacją, niedostępną dla nowej technologii przetwarzania. Jednym ze sposobów wykorzystania tej dokumentacji jest wprowadzenie jej do komputera poprzez klawiaturę, co jednak trwa długo i jest zajęciem bardzo pracochłonnym. Dopiero skonstruowanie skanera pozwoliło na znaczne udoskonalenie systemu automatycznego odczytu dokumentów i wprowadzania zarówno w trybie on-line jak i off-line. Przechowywanie zeskanowanych dokumentów bez żadnej obróbki ma jednak pewne wady. Jedną z nich jest duża wielkość plików (na przykład, zeskanowany z rozdzielczością 300 dpi czarno-biały dokument A4 zajmuje około 1MB). Innym problemem jest trudność przeszukiwania takich dokumentów (szukanie dokumentów o danej tematyce, zawierających konkretne sformułowania, słowa itp.). Jeśli idzie o pierwszą trudność, to wielkość plików można zredukować posługując się metodami kompresji. Nie ułatwi to jednak przeszukiwania dokumentów. Rozdzielenie tekstu i elementów graficznych, a w dalszej kolejności rozpoznanie znaków tekstu oraz analiza elementów graficznych (wydzielanie linii prostych i krzywych [3], rozpoznawanie znanych kształtów [2]) pozwoliłoby na rozwiązanie obydwu wspomnianych powyżej problemów.

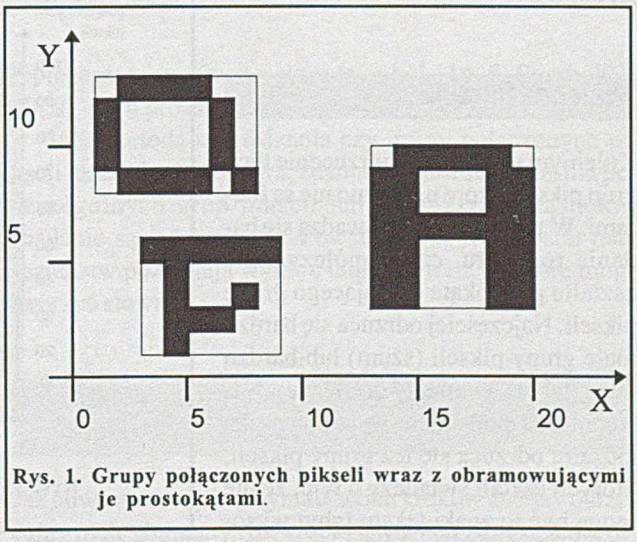
Techniki wydzielania tekstu z dokumentów są rozwijane od lat sześćdziesiątych i początkowo były stosowane do lokalizacji kodów pocztowych, czy cyfr na rachunkach bankowych. Podstawą wielu z tych technik jest informacja kontekstowa na temat typu wydzielanego obszaru, np. tekst tworzy poziome paski na obrazie dokumentu, wszystkie znaki mają ten sam rozmiar. Na podstawie tych informacji tekst może być odróżniany od innych elementów dokumentu.

Sposób separacji tekstu zależy głównie od sposobu tworzenia obrazu dokumentu, tzn. czy to jest obraz rastrowy zawierający odcienie szarości, czy rastrowy obraz binarny, czy wreszcie wektorowy. Opracowane do tej pory metody bazują na własnościach wydzielanych obszarów i działają głównie na dokumentach przechowywanych w postaci mapy bitowej.

Zajmijmy się więc zagadnieniem oddzielania obszarów tekstu od innych obszarów dokumentu zwanych dalej grafiką (rysunki, schematy, fotografie) aby następnie wydzielone obszary można poddawać dalszej obróbce.

## Wydzielanie grup połączonych pikseli

W większości metod analizy dokumentów pierwszym etapem przetwarzania jest wydzielanie grup połączonych pikseli [1], [4], [5]. Szukanie grup połączonych pikseli polega na grupowaniu sąsiadujących ze sobą pikseli o jednakowym kolorze. Wyróżnia się dwa typy sąsiedztwa pikseli: bezpośrednie, jeżeli piksele stykają się bokami, lub pośrednie, jeżeli połączone są ze sobą narożnikami. Niektóre metody analizy przy tworzeniu grup połączonych pikseli posługują się sąsiedztwem bezpośrednim [4], a niektóre pośrednim [9].




Rys. 1. Grupy połączonych pikseli wraz z obramowującymi je prostokątami.

Wynikiem poszukiwania grup połączonych pikseli jest lista, na której najczęściej zapisane są następujące informacje:

- współrzędne prostokąta opisującego grupę połączonych pikseli,
- jego współczynnik kształtu (wysokość do szerokości),
- pole powierzchni prostokąta obramowującego oraz ilość czarnych pikseli wchodzących w skład grupy,
- „ziarno" - czyli piksel grupy, który został wykryty jako pierwszy (pozwala to na odnalezienie na obrazie wejściowym pozostałych pikseli grupy).

| Grupa | Współrzędne | Liczba pikseli czarnych | Pole pow. prostokąta obramowującego | Współczynnik kształtu |
|---|---|---|---|---|
| 1 | (1, 8) - (8, 13) | 15 | 35 | 5/7 |
| 2 | (3, 1) - (9, 6) | 13 | 30 | 5/6 |
| 3 | (13, 3) - (20, 10) | 32 | 49 | 1 |

**Tabela 1. Informacje o grupach połączonych pikseli.**

Przykład prostokątów opisanych na grupach połączonych pikseli oraz związane z nimi informacje pokazano odpowiednio na rysunku 1 i w tabeli 1.

W większości metod analizy dokumentów rozważa się tylko napisy w jednym kolorze (czarne na białym tle). We wszystkich tych metodach szuka się zbiorów połączonych czarnych pikseli. Zdarzają się jednakże obrazy, w których występują zarówno czarne napisy na białym tle, jak i białe- na czarnym. Jest to przypadek bardziej złożony i trudniejszy do analizy, ale znane są już pewne metody analizy takich dokumentów [9].

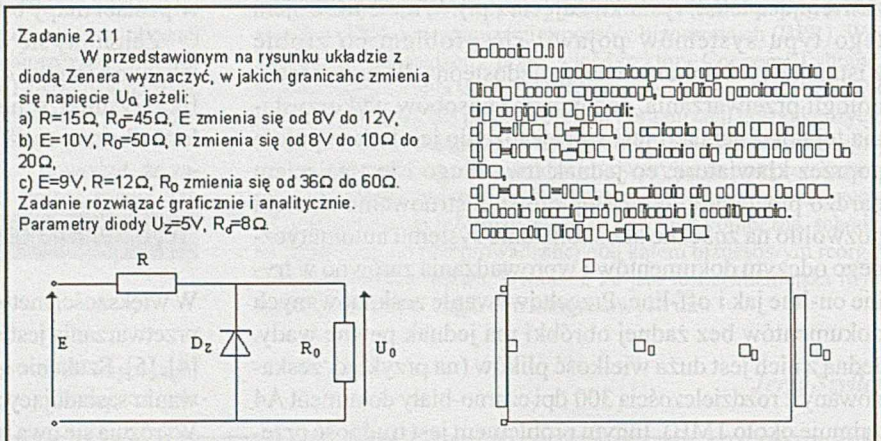
Na rysunku 2 pokazano przykład dokumentu złożonego z tekstu i elementów graficznych i znalezione prostokąty obramowujące wszystkie grupy połączonych pikseli.

## Wstępna filtracja

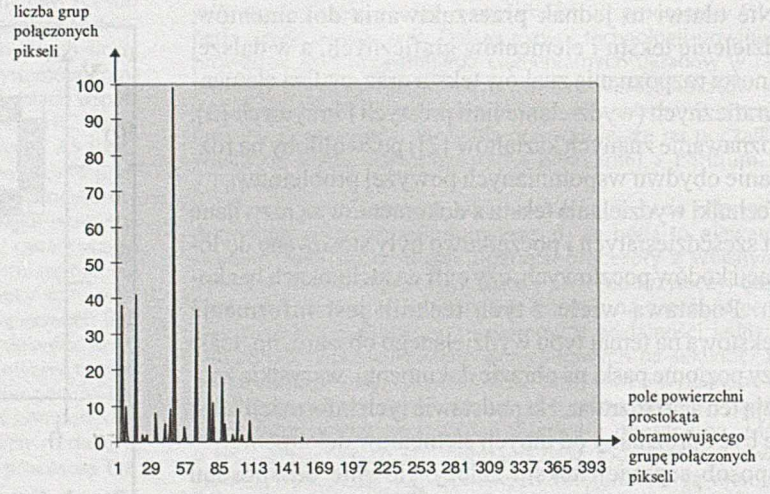
Kolejnym etapem jest odrzucenie tych grup pikseli, które na pewno nie są literami. W tym celu przeprowadza się badanie rozmiaru, czy współczynnika kształtu prostokąta opisującego grupę pikseli. Najczęściej odrzuca się bardzo małe grupy pikseli (szum) lub bardzo duże (tabele, elementy graficzne). Następnie na podstawie współczynnika kształtu odrzuca się też grupy pikseli, których kształt świadczy o tym, że nie mogą być to znaki tekstu (zbyt wysokie lub zbyt szerokie). Na przykład w pracy [1] odrzucono grupy, dla których współczynnik kształtu był mniejszy od 1/20 lub większy od 20/1.

Podczas wstępnej filtracji można wykorzystać z góry posiadane informacje na temat analizowanego dokumentu (ilość znaków tekstu w dokumencie, wielkość stosowanej czcionki). Przydatna jest także wiedza jaką możemy uzyskać analizując histogram wielkości grup połączonych pikseli.

Analizując taki histogram, możemy stwierdzić jakiej wielkości grupy pikseli występowały najczęściej i te najprawdopodobniej odpowiadają znakom. Pozwala to na wykrycie znaków bez względu na wielkość czcionki. W histogramie takim możemy również wykryć wiele małych grup (szum, kropki), jak i pojedyncze duże grupy pikseli stanowiące elementy graficzne Na rysunku 3 pokazano histogram wystąpień grup połączonych pikseli o różnej wielkości dla obrazu z rysunku 2a. Analizując ten histogram łatwo zauważyć, że




**Rys. 2. a) Przykładowy obraz,**
**b) Prostokąty obramowujące grupy połączonych pikseli dla tego obrazu.**




**Rys. 3. Histogram częstości wystąpień grup połączonych pikseli o danej wielkości dla obrazu z rysunku 2a.**

najwięcej grup połączonych pikseli ma wielkość około 43. Wobec tego można założyć, że to są grupy pikseli będące znakami tekstu. Na obrazie występuje także około czterdziestu grup o wielkości 16 oraz 65. Wobec tego, może się zdarzyć (np. jeżeli nie jest powiedziane z góry, że w dokumencie występuje tylko jedna wielkość czcionki), że we wstępnej analizie zostanie założone występowanie na obrazie także tekstu pisanego tymi wielkościami czcionek. Grupy o wielkości 2 do 3, są traktowane jako szum i odrzucane w pierwszej fazie analizy. Jedna grupa o wielkości 156 może być także odrzucona w pierwszej fazie analizy, gdyż najprawdopodobniej jest elementem graficznym.

## Poszukiwanie słów, zdań, paragrafów

Samo wykrycie grup pikseli, których wielkość i kształt mogą sugerować, że są to znaki tekstu nie pozwala jeszcze na stwierdzenie, że tak jest faktycznie. Kolejnym etapem jest więc wykrycie czy grupy połączonych pikseli układają się w słowa, zdania, paragrafy. W tym celu możemy posłużyć się badaniem liniowości ich położenia oraz analizując odległości pomiędzy nimi. Zgodnie z naszymi założeniami znaki tekstu powinny układać się wzdłuż linii, a odległości pomiędzy nimi powinny być dużo mniejsze od ich wysokości.

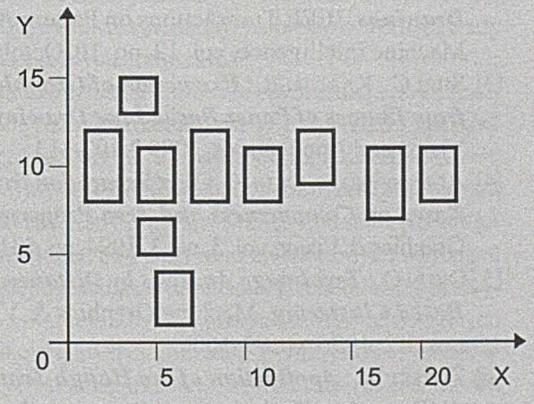
Do badania liniowości (albo do detekcji prostych w obrazie) najczęściej wykorzystuje się transformację Hough'a, która opisana jest wzorem:

$$\rho(\theta) = x\cos(\theta) + y\sin(\theta).$$

Transformacja ta przedstawia relacje pomiędzy punktami $(x, y)$ w przestrzeni obrazów, a punktami (które mogą być traktowane jako wektory) w przestrzeni parametrów $(\rho, \theta)$. Wektor $(\rho, \theta)$ (gdzie $\rho$ jest długością wektora, a $\theta$ jego kątem nachylenia), reprezentuje prostą, która jest do niego prostopadła i przechodzi przez jego koniec. Transformację Hough'a można wykorzystać zarówno do szukania współliniowych grup połączonych pikseli [1], bezpośredniego szukania prostych w obrazie [6], jak i do korygowania „skosów" wynikłych z niedokładnego ułożenia dokumentu na skanerze [8].

Aby znaleźć grupy połączonych pikseli układające się w linię prostą do obliczeń bierzemy współrzędne środków grup. Dla każdej grupy, wyznaczamy jej środek i wyznaczamy wartości $\rho$ dla prostych o poszukiwanych kątach. Wyniki zapisujemy w dwuwymiarowej tablicy (dla dyskretnych wartości $\rho$ i $\theta$). Zapisy do tablicy polegają na zwiększaniu o jeden komórki tablicy, na którą wskazują parametry $\rho$ i $\theta$ obliczone dla środka każdej grupy. Na początku analizy dokumentu tablica wypełniona jest samymi zerami. Po rozważeniu wszystkich środków grup, w tablicy otrzymujemy liczby oznaczające ilość grup, których środki leżą na prostej reprezentowanej przez odpowiedni wektor. Następnie wyszukujemy w tej tablicy komórki o największych wartościach (lokalne maksima), aby znaleźć proste które przechodzą przez największą liczbę grup. Przykładowe ułożenie grup połączonych pikseli oraz współrzędne ich środków przedstawiono na rys. 4. Dwuwymiarowa tablica liczników przedstawiona jest w tabeli 2.

Z tabeli można odczytać, że dla $\rho = 10$ oraz $\theta = 90^0$licznik wynosi 7, a zatem siedem środków grup układa się w prostą reprezentowaną przez wektor (symbol 114 \f „Symbol" \s 10, $\theta$), czyli prostą y = 10.




| X | Y |
|---|---|
| 2 | 10 |
| 4 | 14 |
| 5 | 6 |
| 5 | 9.5 |
| 6 | 2.5 |
| 8 | 10 |
| 11 | 9.5 |
| 14 | 10.5 |
| 18 | 9 |
| 21 | 9.5 |

**Rys. 4. Przykładowe ułożenie grup połączonych pikseli (po lewej) oraz współrzędne środków grup (po prawej).**

Inną metodą sprawdzania czy grupy połączonych pikseli układają się w słowa, zdania, itd. jest szukanie dla poszczególnych grup pikseli ich najbliższych sąsiadów (badanie sąsiedztwa) [9]. Powstające w ten sposób listy sąsiedztw pozwalają wykryć takie sekwencje, które mogą tworzyć słowa.

## Poszukiwanie utraconych elementów tekstu i dodatkowych elementów znaków

Jak łatwo zauważyć w wyniku takich kroków analizy odrzucamy małe elementy tzn. kropki czy znaki zmiękczenia dla

| ρ \ θ | -20° | -10° | 0° | 10° | 20° | 30° | 40° | 50° | 60° | 70° | 80° | 90° | 100° | 110° | 120° |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 |
| 2 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | 3 |
| 4 | 2 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 6 | 0 | 2 | 3 | 4 | 3 | 2 | 1 | 1 | 1 | 0 | 1 | 1 | 3 | 3 | 1 |
| 8 | 1 | 0 | 1 | 0 | 2 | 1 | 2 | 2 | 1 | 1 | 0 | 0 | 4 | 2 | 1 |
| 10 | 1 | 1 | 0 | 1 | 1 | 2 | 1 | 1 | 2 | 2 | 2 | 7 | 1 | 0 | 1 |
| 12 | 0 | 1 | 1 | 1 | 0 | 1 | 2 | 1 | 1 | 2 | 4 | 0 | 0 | 1 | 0 |
| 14 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 2 | 2 | 3 | 2 | 1 | 1 | 0 | 0 |
| 16 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 0 |
| 18 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 20 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 22 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Tabela 2. Interpretacja dyskretnej przestrzeni Hough'a za pomocą tablicy (dla przykładu z rysunku 4).**

polskich znaków. Tracimy też te znaki, które stykają się z dużymi elementami graficznymi lub na przykład w wyniku podkreślenia są połączone w większą grupę. Większość metod pozostawia te elementy jako części elementów graficznych lub odrzuca jako szum [1], [4]. W niektórych jednak przeprowadzana jest próba dołączenia takich elementów do już istniejących słów, zdań czy paragrafów [9]. W tym celu można posłużyć się informacjami zdobytymi podczas szukania elementów tekstu, takimi jak średnia wielkość znaków, odstępy między znakami, odstępy między słowami oraz odstępy między liniami. Wykorzystując te informacje można powiększać granice odnalezionych słów czy zdań i sprawdzać czy w tak powiększonych granicach nie znajdują się inne grupy połączonych pikseli, które mogły by zostać dołączone do już wykrytych elementów tekstu.

❖ ❖ ❖

Wszystkie wspomniane w artykule metody mają pewne ograniczenia oraz przyjmują pewne założenia dotyczące analizowanych dokumentów. Większość z nich została stworzona dla konkretnego typu dokumentów (np. dokumenty techniczne [4], korespondencja firmy [9]). W większości metod gubione są pewne, traktowane jako szum, elementy tekstu (np. kropki nad i). Często fragmenty tekstu stykające się z elementami graficznymi są błędnie klasyfikowane jako elementy graficzne (np. litery dotykające linii tabelki). Z reguły tekst może być tylko w jednym kolorze (albo czarny tekst na białym tle, albo biały na czarnym).

Rozwój tego typu metod polega na próbach ich ulepszania oraz tworzenia nowych metod poprzez łączenie różnych, znanych już algorytmów (lub ich części). Obecnie coraz częściej stosuje się różnego rodzaju filtry - między innymi filtry morfologiczne [7]. Przyszłością tej dziedziny są także systemy eksperckie oraz sieci neuronowe. W metodach takich, system będzie uczył się i zapamiętywał informacje o budowie dokumentów, aby później wykorzystać je do wyboru optymalnego algorytmu.

**Literatura**


[1] Fletcher L. A., Kasturi R.: ***A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images***. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 10, no. 6, November 1988

[2] Kasturi R. , Bow S. T., El-Masri W., Shah J., Gattiker J. R., Mokate U. B.: ***A System for Interpretation of Line Drawings***. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 10, October1990

[3] Shih C., Kasturi R.: ***Extraction of Graphic Primitives from Images of Paper Based Line Drawings***. Machine Vision and Applications, 1989 2:103-113

[4] Ablameyko S., Okun O.: ***Text Separation from Graphics Based on Compactness and Area Properties***. Machine Graphics & Vision vol. 3, no. 3, 1994, pp. 531-541

[5] Okun O.: ***Text Image Analysis by Distance Transform-Based Clustering***. Machine Graphics & Vision vol. 5, nos. 1/2, 1996, pp. 99-100

[6] Żorski W.: ***Application of the Hough Transform with a Clustering Technique to Segmentation of Digital Images***. Machine Graphics & Vision vol. 5, nos. 1/2, 1996, pp. 111-121

[7] Low A.: ***Introductory Computer Vision and Image Processing***. McGraw-Hill Book Company Europe,1991

[8] Srihari S. N., Govindaraju V.: ***Analysis of Textual Images Using the Hough Transform***. Machine Vision and Applications, 1989 2:141-153

[9] Baumann S., Malburg M., Hein H. G., Hoch R., Kieninger T., Kuhn N.: Report ***Document Analysis at DFKI, Part 1: Text Recognition***. German Research Center for Artificial Intelligence (DFKI), Kaiserslauten, FRG.



**Beata Kwidzińska i Witold Malina**
są pracownikami Katedry Technik Programowania Politechniki Gdańskiej.

# Architektury Klient-Serwer

Krzysztof Jeżak

Niewątpliwie przyszłość aplikacji klient-serwer należy do rozproszonych obiektów. Doskonale o tym wiedzą firmy zrzeszone w OMG (m.in. Digital, HP, IBM, SUN) jak i ich konkurent Microsoft. Obie strony mają wiele atutów i na razie trudno przewidzieć, kto wygra. Pewne jest natomiast, że zwycięzca podbije świat i że budując współczesny system informatyczny trzeba uwzględnić rozproszone obiekty.

Wraz z upowszechnieniem komputerów osobistych i wzrostem wymagań użytkownika oraz jako alternatywa dla systemów scentralizowanych, w połowie lat osiemdziesiątych pojawiły się systemy o architekturze klient-serwer. Ewoluowały one od modelu file-serwer poprzez modele dwurzędowe do trójrzędowych. Obecnie jesteśmy świadkami pojawienia się nowego paradygmatu - rozproszonych obiektów. Jakie są wady i zalety poszczególnych architektur?

Architektura to sztuka planowania i tworzenia, a do jej realizacji używa się różnych materiałów. W przypadku systemów klient-serwer „materiały" to: komputery oraz sieć komputerowa, system operacyjny komputerów i sieci, system zarządzania bazą danych, języki i systemy programowania. Podobnie jak w klasycznej architekturze, pod wpływem zwiększonych wymagań użytkowników, powstają nowe materiały i techniki ich wykorzystania, a te z kolei umożliwiają powstanie nowych, lepszych i bardziej funkcjonalnych architektur. W przypadku systemów komputerowych wymagania te to: szybki czas reakcji, możliwość zwiększenia mocy systemu (scalability), niezależność od sprzętu i systemów operacyjnych (portability), modularność, możliwość ponownego użycia istniejących komponentów, scentralizowane zarządzanie bezpieczeństwem i kontrolą, możliwość modyfikowania i adaptowania aplikacji do zmieniających się wymagań użytkownika. Zadaniem architekta jest zaplanowanie systemu, który spełnia wymagania odbiorcy, harmonizuje z otoczeniem (istniejącymi systemami i sprzętem) i w którym każdy detal ma swój cel, a system tworzy spójną całość.

## Definicja

Aplikacja klient-serwer to zbiór połączonych w logiczną i funkcjonalną całość procesów, które są umieszczone na konkretnych komputerach połączonych siecią komputerową. Procesy te dzielimy na usługobiorców (klient) i usługodawców (serwer). Przyjęło się mówić, że komputer, na którym jest umieszczony proces klienta to klient, a komputer z działającym procesem serwera to serwer, nic jednak nie stoi na przeszkodzie, aby na jednym komputerze pracowały oba rodzaje procesów. W dalszej części artykułu będę używać nazw klient i serwer dla oznaczenia procesu lub komputera.

## Komponenty Logiczne

Aby łatwiej zrozumieć różne modele aplikacji klient-serwer, podzielmy ją na komponenty logiczne:

**warstwa prezentacji** to część aplikacji zajmująca się prezentacją danych i konwersacją z użytkownikiem. Można ją podzielić na dwie części:

a. **część usługową**, zajmującą się wyświetlaniem informacji na ekranie i rozpoznawaniem działań użytkownika (naciśnięcie klawisza);
b. **część zarządzającą prezentacją**, która kontroluje konwersację z użytkownikiem, specyfikuje zachowanie aplikacji, gdy użytkownik naciśnie klawisz, wybierze jedną z opcji menu, zamknie lub otworzy okno.

**warstwa logiczna systemu (*business logic*)** to zbiór reguł i algorytmów, które definiują modelowany proces. Może to być algorytm obliczający podatki, oceniający IQ pracownika, czy też zarządzający bankiem lub szpitalem.

**warstwa zarządzająca danymi** dzielimy ją na trzy części:

a. **część konstrukcyjną** - tworzy polecenia, które muszą być wykonane, aby dane zostały zapamiętane w bazie danych. W przypadku relacyjnej bazy danych, jest to zbudowanie poleceń języka SQL;
b. **część wykonującą** - wykonuje wcześniej zbudowane wyrażenia operujące na bazie danych (wykonuje zdania SQL);
c. **część zarządzającą plikami** - wykonuje polecenia systemu operacyjnego zarządzające plikami.

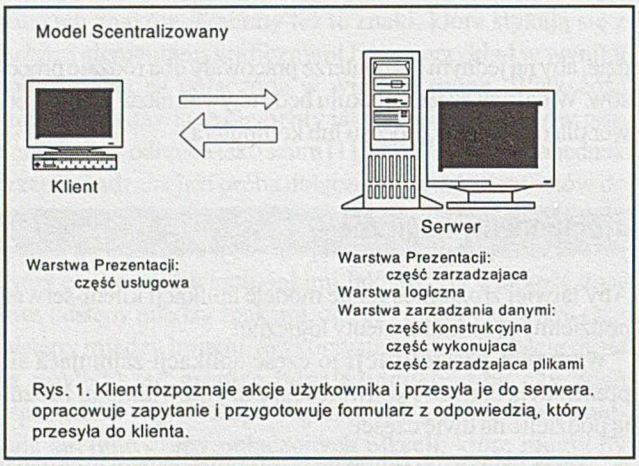
## Modele Fizyczne

Sposób, w jaki warstwy logiczne są umieszczone na konkretnych komputerach definiuje architekturę systemu. Jej ewolu-

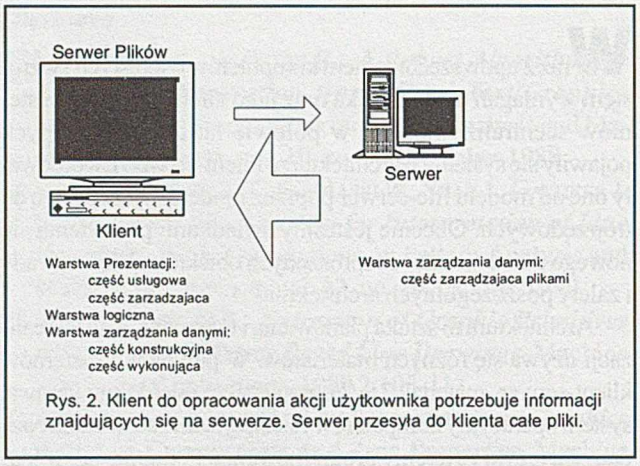
cja spowodowana jest szybkim postępem technicznym, który jest odpowiedzią na nieustannie zwiększające się wymagania użytkowników. Przyjrzyjmy się temu procesowi nieco bliżej.

### *Systemy scentralizowane*

Pierwsze systemy komputerowe były budowane w oparciu o jeden główny komputer. Wszystko było proste, bo cały system pochodził od jednego dostawcy, który dostarczał sprzęt, obsługujących go specjalistów oraz tworzył aplikacje. Jeden komputer obsługiwał niemal cały system, tzn. wszystkie warstwy logiczne oprócz części usługowej, warstwy prezentacji były zaimplementowane na jednym komputerze. Do centralnego komputera były podłączone nieinteligentne (tzn. nie wykonujące żadnych operacji) terminale, których zadaniem było wyświetlanie danych i rozpoznawanie, jaki klawisz został naciśnięty. Dużą zaletą takiego podejścia była prostota architektoniczna oraz łatwe utrzymanie systemu; jeśli coś nie działało, wiadomo było, do kogo dzwonić.




Rys. 1. Klient rozpoznaje akcje użytkownika i przesyła je do serwera. opracowuje zapytanie i przygotowuje formularz z odpowiedzią, który przesyła do klienta.




Rys. 2. Klient do opracowania akcji użytkownika potrzebuje informacji znajdujących się na serwerze. Serwer przesyła do klienta całe pliki.

Wraz z popularnością komputerów PC pojawiły się tysiące użytkowników programujących i wykorzystujących je, dla których systemy scentralizowane były niedostępne. Pojawił się także interfejs graficzny użytkownika (Graphical User Interface), do którego systemy te nie były przystosowane i mogły współpracować z PC, gdy PC emulował terminal albo centralny komputer był zmuszony opracowywać cały ekran graficzny, co powodowało duże obciążenie komputera i sieci komputerowej. Systemy scentralizowane były drogie i mogli sobie na nie pozwolić tylko bogaci klienci, czas tworzenia aplikacji był długi, system nie był skalowalny (trudno było zwiększyć jego moc obliczeniową) i po przekroczeniu pewnej liczby użytkowników, trzeba było kupować nowy, mocniejszy komputer, co często oznaczało budowę aplikacji od nowa. Dlatego w połowie lat osiemdziesiątych, jako alternatywa dla systemów scentralizowanych, pojawiły się systemy otwarte, a wraz z nimi systemy klient-serwer. Otwarte, gdyż każdy komponent systemu mógł pochodzić od innego dostawcy, a co najważniejsze, można je było dowolnie łączyć i dzięki temu budować system dokładnie spełniający wymagania i potrzeby klienta.

Systemy scentralizowane wcale nie muszą oznaczać zastosowania komputerów i technologii z lat siedemdziesiątych. Można je z powodzeniem budować używając współczesnych komputerów i relacyjnych baz danych. Mimo nowszych technologii architektura ta ma podstawowe wady: brak możliwości stopniowego zwiększenia mocy systemu, niemodny sposób prezentacji danych, słaba integracja z innymi systemami, odporność na zmiany i modyfikacje, nieprzenoszalność pomiędzy systemami.

### *Serwer plików*

Nie wszyscy architekturę tę zaliczają do kategorii klient-serwer. Architektura ta dzieli aplikację na dwie części: część klienta zawierającą warstwę prezentacji i logiczną systemu oraz część serwera, zawierającą część zarządzającą danymi. Taki podział rozwiązuje problem interfejsu graficznego (klient sam nim zarządza). System jest dobrze skalowalny, gdyż każdy nowy komputer klienta oznacza dodatkową moc obliczeniową. Wszystkie pliki przechowujące dane znajdują się na wydzielonych komputerach – serwerach, które magazynują i udostępniają zasoby innym komputerom – klientom.

Niestety, architektura ta ma wiele wad: komputery - klienci muszą mieć dosyć dużą moc obliczeniową, gdyż to one wykonują większość przetwarzania. Drugi mankament to sposób pracy z serwerem: aby klient mógł pracować na pliku znajdującym się na serwerze, cały plik musi zostać załadowany do klienta, co powoduje obciążenie sieci komputerowej.

W ten sposób pracowały systemy stworzone przy pomocy takich narządzi, jak Clipper czy FoxPro. Mogły one obsługiwać do około dwunastu klientów pracujących jednocześnie, o ile liczba transakcji i wielkość przekazywanych plików nie była zbyt duża

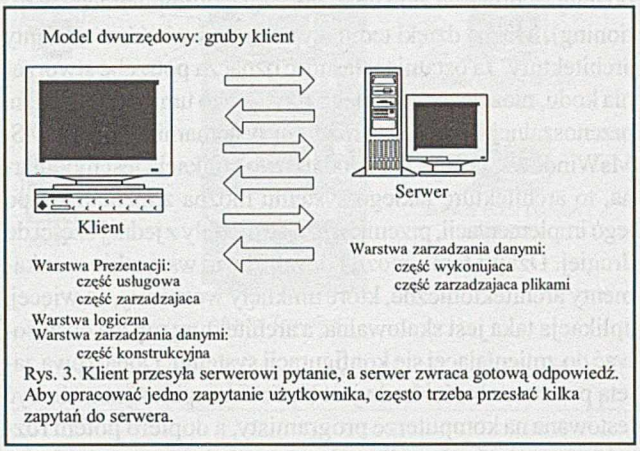
### *Modele dwurzędowe*

W tym modelu cała warstwa prezentacji będzie zawsze rezydowała po stronie klienta, a część wykonawcza i zarządzająca plikami warstwy zarządzania danymi - po stronie serwera. Serwer plików zastąpiono systemem zarządzania bazą danych: zamiast przekazywać całe pliki, wyszukuje on odpowiednie dane i przekazuje tylko odpowiedź. W ten sposób ruch w sieci został poważnie zmniejszony. W zależności od umieszczenia części konstrukcyjnej warstwy zarządzającej danymi oraz warstwy logicznej na kliencie lub serwerze rozróżniamy różne warianty tej architektury:

#### *Zdalne dane - gruby klient*

Część klienta w tym modelu zawiera warstwę prezentacji, warstwę logiczną i część konstrukcyjną warstwy zarządzania danymi. Zaletą tej architektury jest bez wątpienia skalowal-

ność, gdyż podobnie jak w serwerze plików każdy nowy klient zwiększa moc systemu.




Rys. 3. Klient przesyła serwerowi pytanie, a serwer zwraca gotową odpowiedź. Aby opracować jedno zapytanie użytkownika, często trzeba przesłać kilka zapytań do serwera.

Niestety, w przypadku bardziej skomplikowanych aplikacji klient jest bardzo obciążony, a zatem musi mieć dużą moc obliczeniową, co w przypadku większej liczby klientów wiąże się ze sporym wydatkiem. Także liczba zapytań do bazy danych jest duża, zwiększa się zatem ruch w sieci, co jest bardzo niekorzystne, bo sieć jest wąskim gardłem każdej aplikacji klient-serwer.

Na wydajność systemu ma także wpływ sposób, w jaki klient przekazuje zapytanie bazie danych. Obecnie powszechnie używa się relacyjnych baz danych i systemów zarządzania relacyjną bazą danych jako serwera. Ponieważ logika aplikacji znajduje się na kliencie, zatem żądania do serwera (zdania SQL) przygotowywane są na kliencie. Istnieją dwie metody przekazywania tych żądań. Pierwsza to zanurzony SQL (embeded SQL) - zdania SQL są zanurzone w innym języku i aplikacja wymaga prekompilacji (zdania SQL zostaną przetłumaczone na procedury języka) nim zostanie skompilowana. Druga metoda to użycie ODBC (Open Database Connectivity - model zdefiniowany przez Microsoft) lub CLI (Call Level Interface - model zdefiniowany przez porozumienie X/Open). Przekazują one zdanie SQL do bazy danych, gdzie zostanie ono skompilowane w momencie wykonania (dynamic SQL). Ten drugi sposób jest wprawdzie niezależny od bazy danych (zajmują się tym sterowniki ODBC lub CLI) i prostszy, lecz zdania SQL są kompilowane podczas wykonania programu, co wydłuża czas opracowania zapytania.

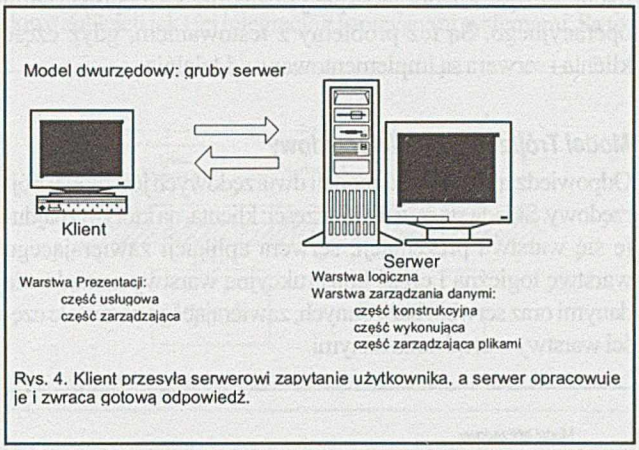
Poza tym, ponieważ logika aplikacji znajduje się w części klienta, każda jej zmiana zmusza do ponownej instalacji części klienta. Może to stanowić problem przy dużej liczbie klientów szczególnie, gdy każdy z nich ma inną konfigurację. Wymusza to także używanie jednego systemu operacyjnego na komputerach klienta albo też aplikację trzeba zaimplementować na różne systemy operacyjne.

Systemy takie buduje się używając systemu Visual Basic (lub PowerBuilder, Delphi), ODBC i bazy danych, systemu Progress i innych.

***Zdalna prezentacja danych - gruby server***

W tym modelu część klienta zawiera tylko warstwę prezentacji, natomiast warstwy logiczna i zarządzania danymi znajdują się na serwerze, dzięki czemu ruch w sieci jest jeszcze bardziej ograniczony.

Jest to możliwe dzięki uprzednio skompilowanym procedurom bazy danych (stored procedures), pisanych w rozszerzonym SQL (np. Sybase: Transact-SQL, Oracle: PL/SQL), które są wykonywane na serwerze. Zamiast wysyłać całe zdanie SQL, klient przekazuje po prostu nazwę i parametry procedury. Procedura taka może wykonać wiele różnych zapytań do bazy danych, zanim ostatecznie przekaże odpowiedź klientowi, zmniejszając tym samym ruch w sieci. Administrowanie taką aplikacją staje się łatwiejsze, gdyż duża jej część jest scentralizowana. Aplikacje budowane w oparciu o tą architekturę z powodzeniem mogą obsłużyć do stu klientów pracujących jednoczenie.




Rys. 4. Klient przesyła serwerowi zapytanie użytkownika, a serwer opracowuje je i zwraca gotową odpowiedź.

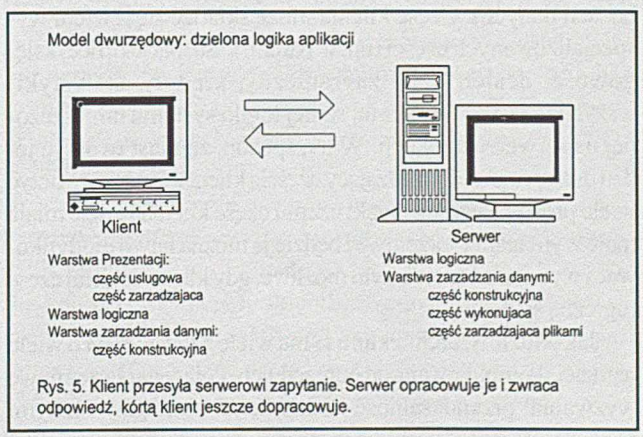
Gdy umieszczamy logikę aplikacji na serwerze, należy pamiętać o tym, że jeśli procedury są zbyt skomplikowane, to serwer będzie „wąskim gardłem" takiego systemu, a sam system będzie słabo skalowalny. Innym problemem są serwery bazy danych. Aby klient mógł z nim nawiązać kontakt, musi najpierw otworzyć sesję, których jest tylko ograniczona ilość. Musimy też pamiętać, że umieszczając logikę aplikacji na serwerze bazy danych, uzależniamy się od konkretnej jej implementacji.

Systemy takie można budować używając m.in. PowerBuilder lub VB i bazy danych, Progressa, narządzi Oracla, Informixa, Ingresa i wielu innych.

***Dzielona logika aplikacji***

Umiejscowienie części logicznej aplikacji na serwerze nadmiernie go obciąża, przez co staje się on słabym punktem systemu, a umieszczenie jej na kliencie powoduje obciążenie sieci i zmusza do zainwestowania sporych sum pieniędzy w klientów o dużej mocy obliczeniowej. Dlatego właśnie stosuje się rozwiązania mieszane: logikę aplikacji można umieścić zarówno na kliencie, jak i na serwerze.

Oprócz wymienionych już wcześniej ograniczeń wszystkie modele dwurzędowe mają problemy z integracją baz danych pochodzących od różnych producentów i przenoszalnością

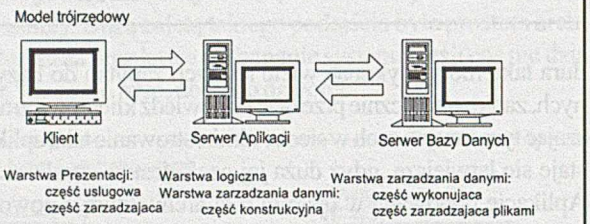



Rys. 5. Klient przesyła serwerowi zapytanie. Serwer opracowuje je i zwraca odpowiedź, kórtą klient jeszcze dopracowuje.

stworzonego kodu między systemami, gdyż procedury bazy danych pisane są w języku specyficznym dla każdej bazy danych, a część klienta jest specyficzna dla każdego systemu operacyjnego. Są też problemy z testowaniem, gdyż część klienta i serwera są implementowane oddzielnie.

### *Model Trójrzędowy /Wielorzędowy*

Odpowiedzią na bolączki modeli dwurzędowych jest model trójrzędowy. Składa się on z trzech części: klienta, na którym znajduje się warstwa prezentacji, serwera aplikacji zawierającego warstwę logiczną i część konstrukcyjną warstwy zarządzania danymi oraz serwera bazy danych, zawierającego pozostałe części warstwy zarządzania danymi.




Rys. 6. Klient przesyła serwerowi zapytanie użytkownika, serwer aplikacji opracowując odpowiedź zwraca się do serwera bazy danych o dane potrzebne do opracowania odpowiedzi.

Umieszczenie warstwy logicznej na serwerze aplikacji powoduje, że jest ona niezależna od konkretnej bazy danych i implementacji procedur bazy danych, a tym samym kod jest łatwiej przenoszalny pomiędzy systemami. Logika aplikacji, którą trudno jest całkowicie umieścić w części klienta lub serwera, teraz znajduje się na oddzielnym serwerze aplikacji. Można umieścić kilka serwerów aplikacji (replikacja), tym samym zwiększając moc obliczeniową systemu. Replikacja powoduje także zwiększenie niezawodności systemu, gdyż mimo awarii jednego z serwerów, cała aplikacja może nadal pracować. Serwer aplikacji można umieścić blisko albo nawet na tym samym komputerze, na którym jest umieszczony serwer bazy danych, zmniejszając tym samym ruch w sieci. Podział aplikacji na trzy części zwiększa modularność systemu, a poszczególne jego części można rozmieszczać w dowolny sposób (serwer aplikacji razem z serwerem bazy danych lub na oddzielnym komputerze). Dane mogą się znajdować w różnych niejednorodnych bazach danych, a serwer aplikacji może korzystać z nich wszystkich (np. z reguły obrazy są przechowywane jako pliki podczas, gdy dane strukturalne przechowywane są w relacyjnych bazach danych). Część klienta może składać się z wielu wyspecjalizowanych części (np. w banku: kasa, back-office, księgowość, dealer, dział zagraniczny, kredyty, statystyki) wszystkie korzystające z tej samej logiki systemu umieszczonej na serwerze aplikacji. W ten sposób, zamiast tworzyć jeden duży program zarządzający częścią klienta, można stworzyć wiele prostych części, dzięki czemu część klienta będzie miała małe wymagania sprzętowe i będzie ją można łatwo modyfikować i wymieniać (co nie było możliwe, gdy klient zawierał część logiczną systemu).

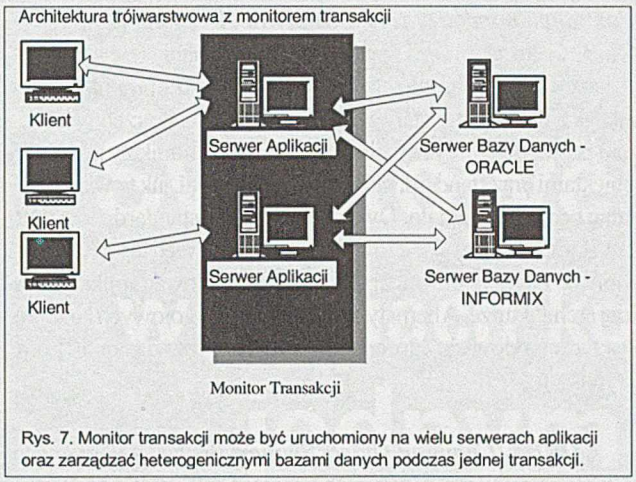
Jak widzimy, architektura ta ma wiele zalet, ale jest o wiele bardziej skomplikowana od poprzednich. Pojawiają się też nowe wyzwania: przenoszalność kodu klienta pomiędzy różnymi systemami (MsWindows, OS/2, Macintosh), współpraca systemów (interoperability) oraz dzielenie aplikacji na części, które później są umieszczane na konkretnych komputerach (partitioning). Można dzięki temu wypróbowywać różne warianty architektury. Ta ostatnia własność oznacza potrzebę stworzenia kodu, niezależnego od jego fizycznego umieszczenia, tzn. przenoszalnego pomiędzy różnymi systemami (UNIX, VMS, MsWindows, OS/2). Jeśli dodatkowo aplikacja jest modularna, to architekturę takiego systemu można zmieniać już po jego implementacji, przemieszczając moduły z jednej części do drugiej. Dzięki temu można doszlifować wszystkie mankamenty architektoniczne, które umknęły wcześniej, co więcej, aplikacja taka jest skalowalna, a architekturę można dostosować do zmieniającej się konfiguracji systemu. Dodatkową zaletą przenoszalności kodu jest to, że cała aplikacja może być testowana na komputerze programisty, a dopiero potem rozmieszczona (deployed) na konkretnych komputerach, co z pewnością pozwoli przyspieszyć testowanie. Z reguły przenoszalność kodu uzyskuje się poprzez wygenerowanie kodu w języku niskiego poziomu, wspólnego dla wszystkich platform (C, C++). Systemy umożliwiające taką pracę to Forte i Dynasty.

Słabą stroną tej architektury jest serwer bazy danych. Wprawdzie został on poważnie odciążony, lecz w przypadku dużego ruchu staje się wąskim gardłem. Jest to spowodowane tym, że każdy klient łącząc się z serwerem, angażuje na nim pewną ilość zasobów (pamięć, jeden lub dwa procesy lub w nowszych systemach wątki, otwiera kilka plików), które są wykorzystywane tylko, gdy klient zażąda wykonania usługi. Jeśli zrobi to tysiąc klientów, każdy system będzie mieć poważne kłopoty.

Serwer aplikacji może przyjąć różną postać w zależności od jej wymagań. Może to być oddzielny program lub programy, kolejny system zarządzania bazą danych, na którym będą wykonywane tylko procedury bazy danych (architektura proponowana przez wielu producentów baz danych), może to być monitor transakcji lub też obiektowa baza danych.

### *Monitor Transakcji (MT)*

Rozwiązaniem problemu ograniczonej ilości sesji bazy danych jest zastosowanie Monitora Transakcji (MT) jako serwera aplikacji, który może zapewnić sprawne działanie systemu, do którego jest podłączonych tysiąc lub więcej klientów. Wykorzystuje on fakt, że tylko niewielka ilość klientów potrzebuje wykonania danej usługi jednocześnie i uruchamia znacznie mniejszą ilość połączeń do bazy danych, niż jest w rzeczywistości klientów. Klient komunikuje się z bazą danych poprzez monitor transakcji: gdy otwiera sesję łączy się z procesem monitora transakcji, który w odpowiednim momencie zażąda usługi od serwera bazy danych. Każdy z procesów MT może uruchomić kilka wątków i może obsłużyć kilku klientów. Aby móc obsłużyć dużą liczbę transakcji jednocześnie, MT zbiera zlecenia, a następnie kolejno je wykonuje. Jeśli ilość procesów MT jest zbyt mała, uruchamiane są nowe procesy (load balancing). Monitor Transakcji może być uruchomiony na kilku komputerach, zapewniając tym samym gładkie zwiększanie mocy systemu. Komunikacja z klientem jest asynchroniczna. Klient po dostarczeniu zapytania do MT dalej się nim nie zajmuje i może wykonać inną pracę, a MT bierze całą odpowiedzialność za obsługę tego zapytania.




Rys. 7. Monitor transakcji może być uruchomiony na wielu serwerach aplikacji oraz zarządzać heterogenicznymi bazami danych podczas jednej transakcji.

MT zapewnia dużą niezawodność systemu. Jeśli stwierdzi, że jeden z serwerów przestał działać, wtedy będzie starał się go uruchomić, a jeśli się to nie uda, może skierować ruch w sieci do innego, działającego serwera.

Inną zaletą MT jest to, że używając protokołu two-phase commit, może on zarządzać wieloma różnymi źródłami danych podczas jednej transakcji (bazy danych pochodzące od różnych producentów, nietypowe źródła danych, np. czytnik kodu kreskowego mogą być wykorzystane na równi z bazami danych), przypisywać priorytet transakcjom. Niektóre MT oferują bardziej zaawansowane techniki transakcyjne (transakcje zanurzone, niedostępne na razie w systemach zarządzania danymi).

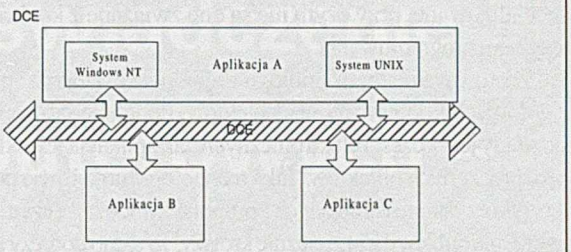
Ceną za używanie tej technologii jest skomplikowane środowisko programowe, bardziej skomplikowany projekt aplikacji, który musi uwzględniać MT, konieczność poznania nowych technik i programowanie w językach niskiego poziomu (C, COBOL).

***Obiektowa Baza Danych***

Bazy Danych przechowują często dane w formie niewygodnej do użycia, szczególnie, gdy używa się obiektowego języka programowania. Opłacalne staje się wówczas użycie obiektowej bazy danych, która przechowuje dane w postaci obiektów, co znacznie skraca czas reakcji systemu. Niestety, systemy zarządzania obiektową bazą danych nie pozwalają na razie zarządzać danymi tak dobrze, jak systemy zarządzania relacyjnymi bazami danych, można jednak je wykorzystać jako pamięć podręczną (cache).

## Rozproszone środowisko komputerowe

Aplikacja klient-serwer jest często rozmieszczona na różnorodnym sprzęcie. Dzięki temu można dobrać sprzęt do wymagań systemu, pojawia się jednak problem współdziałania produktów wykonanych w różnych technologiach. Rozproszone środowisko komputerowe DCE (Distributed Computing Environment), zdefiniowane przez OSF (Open Software Foundation), to technologia umożliwiająca współdziałanie różnych systemów operacyjnych i technologii sieciowych. Warto o tej technologii wspomnieć, gdyż niemal wszyscy producenci oprogramowania ją podtrzymują, zatem oferuje niezależność od systemu operacyjnego, sieci komputerowej i sprzedawcy. Jest uniwersalnym middleware, który jest o wiele łatwiejszy w użyciu, niż inne podobne technologie, a dodatkowo oferuje szereg usług umożliwiających rozproszenie danych i procesów, co ułatwia zarówno budowę aplikacji jak i jej integrację z istniejącymi systemami. Są to:




Rys. 8. DCE umożliwia integrację różnych systemów.

- zdalne wołanie procedur (RPC - Remote Procedure Call): komunikacja pomiędzy komponentami systemu znajdującymi się na różnych systemach;
- usługa zapewniająca bezpieczeństwo (security service): uwiarygodnianie użytkowników, zezwalanie na korzystanie z zasobów, kontrola integralności komunikatów (czy odebrany komunikat nie został zmieniony podczas transmisji), szyfrowanie;
- usługa zarządzająca spisem usług (directory service): zapewnia jednoznaczne nazewnictwo dostępnych usług i pozwala klientowi znaleźć serwera;
- usługa czasu (time service): synchronizuje czas w sieci;
- usługa wątków (*thread service*): tworzy i zarządza wątkami;
- usługa zarządzająca rozproszonymi plikami (*distributed file service*).

### Middleware

to część aplikacji pomiędzy sprzętem a oprogramowaniem. Oferuje zestaw przezroczystych dla programisty usług umożliwiających połączenie różnych części aplikacji w całość, niezależnie od ich umiejscowienia, sprzętu i systemu operacyjnego, formatu danych, języków, w których usługi są zrealizowane, protokołów sieciowych. Middleware to zbiór wielu różnych technologii, które w sumie muszą oferować następujące usługi:

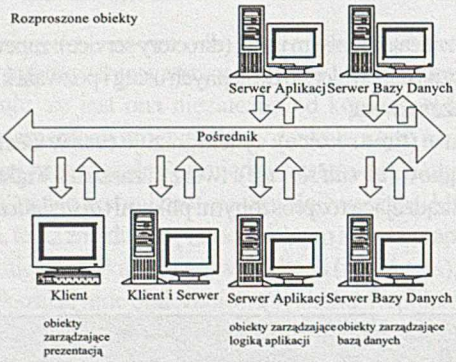
- usługi komunikacyjne: DCE powinien współpracować z najważniejszymi protokołami komunikacyjnymi (IPX/SPX, TCP/IP), łączyć je i tłumaczyć komunikaty pomiędzy nimi;
- usługi dostępu do danych i połączenia: powinien oferować różne standardy połączenia klienta i serwera (ODBC, CLI), umożliwiać dostąp do różnych struktur danych (pliki, tablice w relacyjnych bazach danych), oferować różne sposoby przekazywania komunikatów pomiędzy procesami (RPC, MOM, ORB);
- usługi zarządzania zasobami: zarządzanie wątkami, kolejkowanie zadań, równoważenie obciążenia (load balancing);

Inne dodatkowe usługi, to: usługi bezpieczeństwa (ujednolicone zarządzanie bezpieczeństwem), transakcyjne (potwierdzenia transakcji i regeneracja danych po awarii w rozproszonym środowisku), usługi nadzorujące i zarządzające systemem.

### *Obiekty Rozproszone*

Obiekty rozproszone doskonale pasują do modelu klient-serwer, gdyż łącząc dane i metody stają się samowystarczalne i mogą być umieszczone w dowolnym miejscu rozproszonej aplikacji. Można je łączyć w większą całość lub mogą być wyspecjalizowane, przy czym nie są one związane z konkretnym językiem programowania.

Zastosowanie tej technologii umożliwia składanie aplikacji z gotowych komponentów - obiektów. Podobnie jak komputer składany jest z części elektronicznych, tak aplikacja jest składana z rozproszonych obiektów. Taka technologia umożliwia ponowne wykorzystanie istniejących komponentów, dzięki czemu czas tworzenia aplikacji jest znacznie krótszy. To samo dotyczy poprawy lub ulepszenia aplikacji - dodanie nowego elementu lub zmiana działania istniejącej monolitycznej aplikacji jest trudna i czasochłonna, gdyż wprowadzenie nowych założeń pociąga za sobą zmianę całości. Dzięki komponentom, zmiana ograniczy się tylko do konkretnych elementów. Co więcej, komponenty można dowolnie umieszczać na dowolnych komputerach, dzięki czemu można łatwo zwiększać moc obliczeniową systemu, a tym samym dostosować go do wymagań użytkownika .




Rys. 9. Aplikacja składa się z obiektów pełniących różne funkcje, komunikujących się poprzez pośrednika.

Technologia ta jest prawdziwą rewolucją w informatyce. Komponenty to swoistego rodzaju klocki, które będą rozwiązywały konkretne zadania i będą mogły być łączone w nieprzewidziane przez twórcę kombinacje. Zapewne pojawią się firmy, które będą sprzedawały gotowe komponenty. Takie komercyjne superkomponenty będą musiały być bezpieczne, posiadać licencję, wersje, zarządzać zasobami, posiadać możliwość obsługi i tworzenia zdarzeń, być transakcyjne, trwałe (persistent), samoinstalujące się, dostosowujące się do środowiska itd.

Pojawią się szkielety aplikacji (Framework) oferujące gotową architekturę systemu, którą będzie można dowolnie rozszerzać i dostosowywać do własnych potrzeb. Podział aplikacji na trzy części oczywiście nadal będzie obowiązywać z tym, że w każdej z części pojawią się obiekty zamiast procesów: obiekty zarządzające prezentacją, logiką aplikacji i bazą danych. Nie będzie już wyraźnego podziału na klienta i serwer: obiekty prezentacji będą przekazywały swoje zapytania do pośrednika, który po znalezieniu odpowiedniego obiektu zarządzającego logiką aplikacji przekaże mu zapytanie. Obiekty będą mogły być umieszczone na dowolnym komputerze tej aplikacji i podział na komputery klienta i serwera zniknie.

### *Pośrednik pomiędzy rozproszonymi obiektami (ORB)*

Obiekty jako takie nie mogą istnieć w odosobnieniu - muszą współdziałać niezależnie od rodzaju sieci, systemu komputerowego i języka, w którym zostały zaimplementowane. Potrzebny jest zatem pośrednik, który będzie przekazywał komunikaty pomiędzy obiektami oraz standard, który by specyfikował, jak to współdziałanie będzie wyglądało. Dwa najważniejsze standardy to: CORBA i DCOM. Używając ich na przykład w zestawach: Java+CORBA lub VB+DCOM, można z powodzeniem budować aplikacje o takiej architekturze. Alternatywą jest używanie gotowych zintegrowanych środowisk, które oferują własne rozwiązania (Forte, Dynasty).

### *CORBA (Common Object Request Broker Architecture)*

Model CORBA zdefiniowany przez OMG (Object Management Group), podobnie jak DCE ma na celu integrację aplikacji klient-serwer w niejednorodnym rozproszonym środowisku komputerowym. Jest on oferowany na wielu platformach i podobnie jak DCE należy do technologii middleware. Oba używają języka specyfikacji IDL (interface definition language) do definiowania dostępnych usług, lecz CORBA jest zorientowana na obiektowe techniki programowania podczas gdy DCE jest zorientowany na programowanie proceduralne.

CORBA definiuje obiekty i sposób, w jaki ze sobą współdziałają. Sposób, w jaki są one zaimplementowane czy pod jakim systemem działają jest przezroczysty, a całe środowisko jest zarządzane przez CORBA. CORBA definiuje bogaty zestaw usług (transakcyjne, bezpieczeństwa, trwałości etc.), które mogą być używane przez dowolny obiekt. Implementacja konkretnych obiektów jest napisana w jednym z języków, który oferuje wiązania do CORBA (C, C++, Ada, Smalltalk, Java, COBOL, Objective C).

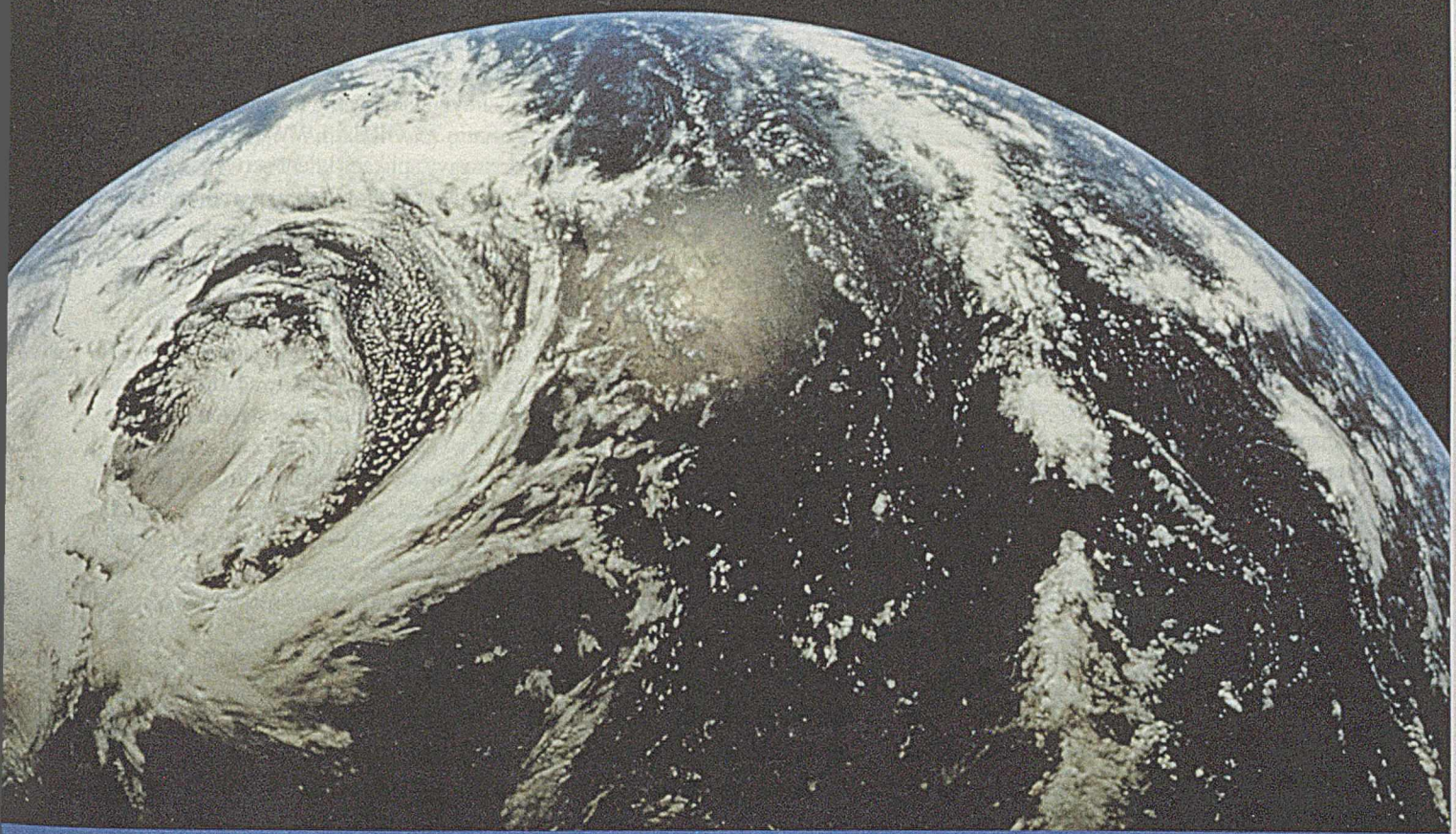
CORBA składa się z:

- ORB - pośrednika;
- Wspólnych Obiektów Usługowych (Common Object Services);
- Wspólnych Udogodnień (Common Facilities);
- Obiektów Aplikacji (Application Objects).

CORBA oferuje statyczne i dynamiczne wołanie metod, dzięki czemu aplikację można zmieniać i ulepszać podczas jej działania. Aplikacja może być pisana w różnych językach, a użytkownik może dowolnie zmieniać umiejscowienie komponentów na konkretnych komputerach. Ta sama aplikacja może działać w trybie jednostanowiskowym lub w środowisku rozproszonym.

CORBA jest bardzo ogólnym modelem, niezależnym od języka i modelu obiektowego. Dzięki temu może łączyć komponenty napisane w proceduralnych lub obiektowych językach programowania. W ten sposób główny cel, jakim jest współdziałanie komponentów i przenoszalność kodu został osiągnięty, ale jednocześnie niemożliwe stało się wykorzystanie własności specyficznych dla konkretnego języka (przekazywanie obiektów pomiędzy komponentami, przekazywanie asynchronicznej odpowiedzi, rozgłaszanie komunikatów i inne). Dlatego CORBA została zaimplementowana przez wielu producentów (Digital: ObjectBroker, IBM: SOM, SUN: JOE, HP: ORB Plus, Expertsoft: PowerBroker, Iona: Orbix to tylko najważniejsze), oferujących przeróżne rozszerzenia. W ten sposób ponownie pojawił się problem, który próbowano rozwiązać:

# Światowe doświadczenie w sieciach komputerowych – jednym słowem...

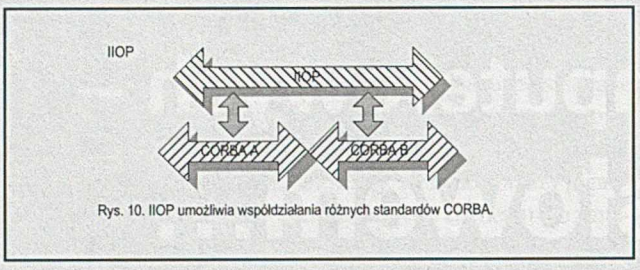


Sun Microsystems. Synonim światowego doświadczenia w dziedzinie sieci. Od początku funkcje sieciowe były wbudowane w każdy z naszych systemów. Dzięki gamie nowatorskich rozwiązań - od aplikacji Java, poprzez serwery i wydajne mikroprocesory, do środowiska operacyjnego Solaris - potrafimy zaspokoić nawet najbardziej różnorodne potrzeby informatyczne przedsiębiorstwa. Współpracujemy z partnerskimi firmami, dobrze znającymi specyfikę polskiego rynku. Sun jest wszędzie tam, gdzie sukces Twojej firmy zależy od przetwarzania informacji. Więcej informacji znajdziesz w Internecie pod adresem: http://www.sun.com. **DOPIERO SIEĆ TO KOMPUTER.™**

Sun microsystems

SUN MICROSYSTEMS POLAND, Warszawa, ul. Hankiewicza 2, tel. 658 45 35, fax 658 40 60, e-mail: info@Poland.Sun.COM

współdziałanie - tym razem różnych standardów CORBA. Tak narodził się protokół IIOP (Internet Inter-ORB Protocol), definiujący sposób porozumiewania się różnych standardów CORBA oraz ESIOP (Environment Specific Inter-ORB Protocol), który definiuje sposób komunikacji CORBA z innymi technikami middleware (DCE).



Rys. 10. IIOP umożliwia współdziałania różnych standardów CORBA.

***DCOM (Distributed Component Object Model)***

To rozszerzenie OLE, będące odpowiedzią Microsoftu na standard CORBA, do którego jest bardzo podobny, ale uboższy: nie oferuje dziedziczenia (zamiast tego oferuje wielokrotne złącze (*interface*), nie jest otwarty na inne platformy poza Windows NT4.0 i następcę Windows'95 - Cairo.

## Uniwersalny Klient

Wprowadzenie komponentów burzy dotychczasowe pojęcie aplikacji jako jednej całości: pojawia się wiele małych i prostych obiektów współpracujących między sobą. Własności tych obiektów i sposób ich współpracy definiuje teraz aplikacją. Jak powinna wyglądać część prezentacji takiej aplikacji? Najlepszą metodą jest umożliwienie użytkownikowi obejrzenie i wykonanie pewnych operacji na niektórych obiektach. Oczywiście, każdy z obiektów będzie posiadał inne własności (atrybuty, metody, stałe etc.) i będzie wymagał innego rodzaju prezentacji (liczby, tekst, grafika, animacja, dźwięk), ale szkielet aplikacji klienta może być taki sam dla każdej aplikacji. Serwer będzie przesyłał dokument (zbiór obiektów) do klienta, który będzie go prezentował. Pojawienie się nowego typu obiektu nie będzie nastręczało trudności w jego prezentacji, gdyż każdy z obiektów będzie zawierał metody nim zarządzające. Taki uniwersalny prezenter dokumentów musi zawierać protokół, za pomocą którego będzie się komunikował z obiektami. Protokół ten musi także umożliwiać wymianę komunikatów pomiędzy obiektami, które „nic o sobie nie wiedzą".

Dwa najważniejsze standardy zarządzające prezentacją złożonych dokumentów to OpenDoc, promowany przez OMG, (rozszerza standard CORBA na część prezentacji) i OLE, promowany przez Microsoft, (współpracujący z DCOM). Funkcjonalnie oba standardy są bardzo podobne: oba umożliwiają współdziałanie obiektów, definiują reguły dla wizualnych komponentów (np. w jaki sposób dzielone jest wspólne okno), oferują automatyzację. OpenDoc dodatkowo oferuje prezentację dokumentów o nieregularnych kształtach oraz możliwość bezpośredniej edycji dowolnego komponentu nieza- leżnie jak głęboko jest on zanurzony, bez konieczności edycji wszystkich dokumentów go otaczających („*in-place*" *editing*).

## Internet

Na zakończenie warto wspomnieć o największych aplikacjach klient-serwer, jakie zostały stworzone, które zawierają ogromną ilość serwerów i jeszcze większą liczbę klientów i rosną w ogromnym tempie. Są to oczywiście aplikacje internetowe takie jak Gopher czy WWW. Tej drugiej warto się przyjrzeć bliżej, gdyż ostatnio robi zawrotną karierę.

Pierwsza generacja aplikacji WWW to statyczne dokumenty hiperkekstowe, które są ładowane i wyświetlane na uniwersalnym kliencie - przeglądarce, która może wyświetlać różne rodzaje dokumentów (tekst, grafika, dźwięk, video). Następnie, dzięki protokołowi CGI pojawiła się możliwość opracowywania odpowiedzi na serwerze. Oba podejścia są bardzo prymitywne w porównaniu z architekturami omówionymi wcześniej: dokumenty mogą być tylko przeglądane, brak jakiegokolwiek bezpieczeństwa czy transakcyjności.

Następna generacja to aktywny klient z ładowalnymi komponentami (applets), umożliwiający znacznie lepsze współdziałanie użytkownika z aplikacją. Umożliwił to język Java i inne języki, które mogą wykonać dynamicznie załadowany kod (np. Tcl). Kod ten jest interpretowany na maszynie klienta pod nadzorem przeglądarki. Jest niezależny od systemu operacyjnego klienta i nie musi być instalowany (sam się ładuje). Co więcej, Java w połączeniu z CORBA i WWW tworzy nową jakość: model trójwarstwowy aplikacji klient-serwer, w którym warstwę prezentacji (klient) tworzą applety Java (komunikujące się z CORBA poprzez Java ORB), obiekty odpowiedzialne za logikę aplikacji (serwer aplikacji) i serwer bazy danych w trzeciej warstwie łączy w całość CORBA.

Niestety applety Java, składające się na część klienta, nie mają możliwości komunikowania się ze sobą (chyba że poprzez CORBA), nie można nimi manipulować ani zanurzać w nich innych komponentów lub appletów. Jest to niewątpliwa niedogodność, która jest wyeliminowana przez takie technologie jak Cyberdog. Zaadoptowała ona technologię OpenDoc do Internetu, tworząc w ten sposób przeglądarkę rozproszonych obiektów żyjących w Internecie, które dzięki tej technologii mogą współdziałać. Komunikacja z serwerem odbywać się będzie poprzez CORBA. Podobną technologię oferuje Microsoft w oparciu o ActiveX, DocObjects i DCOM.

**Literatura:**

[1] Brando T.: ***Comparing DCE and CORBA***, March 1995. MITRE Document MP95D-93
[2] Carone T.E.: ***Middleware and Three-Tier Client/Server Development***, November 1996. Dr.Dobbs
[3] Chen G.: ***Distributed Transaction Processing Standards and Their Applications***, 1995. Computer Standards and Interfaces
[4] Edelstein H.: ***Unraveling Client/Server Architectures***, May 1994. DBMS
[5] Orfali R., Harkey D., Edwards J.: ***The Essential Client/Server Survival Guide Second Edition***, 1996. John Wiley & Sons, Inc.
[6] (Quinlan T.: ***The Second Generation of Client/Server***: May 1995. Database Programming and Design
[7] Resnick R.: ***Intergalactic Distributed Objects***, January 1997. Dr.Dobbs
[8] Silver J.: ***3 -Tier with no tears***, 1996. Oracle Corporation Canada Inc.

**Krzysztof Jeżak** jest pracownikiem Sage Solutions Inc. w San Francisco.

# Baza Danych
## Fizykochemicznych i Termofizycznych

Marcin Gorawski, Jacek Frączek

Baza danych zawiera informacje o eksperymentach przeprowadzonych na około 100,000 substancji chemicznych, w tym także na układach dwu– i wieloskładnikowych. Eksperymenty te obejmują swym zakresem około 100 własności fizykochemicznych i termofizycznych substancji.

Zgromadzone dotychczas zasoby danych stanowią jeden z największych zbiorów informacji fizykochemicznych na świecie. Baza danych jest przedmiotem eksportu oraz porozumień, umożliwiających jej wykorzystanie w najpoważniejszych systemach symulacji komputerowej procesów przemysłowych (CHEMCAD, ASPEN, PRO-II). Ze zbiorami danych współpracuje obszerne oprogramowanie służące do wyszukiwania danych, ich prezentacji tekstowej i graficznej, a także do wyliczania danych przewidywanych, gdy baza nie zawiera odpowiednich danych, itp. Źródłem danych są publikacje naukowe.

Z uwagi na ogromną wartość i ilość zgromadzonych danych, zdecydowano się na ich przeniesienie do nowego systemu Bazy Danych Fizykochemicznych i Termofizycznych (**Bazy DaFiT**), opartego o system zarządzania bazą danych Oracle7.

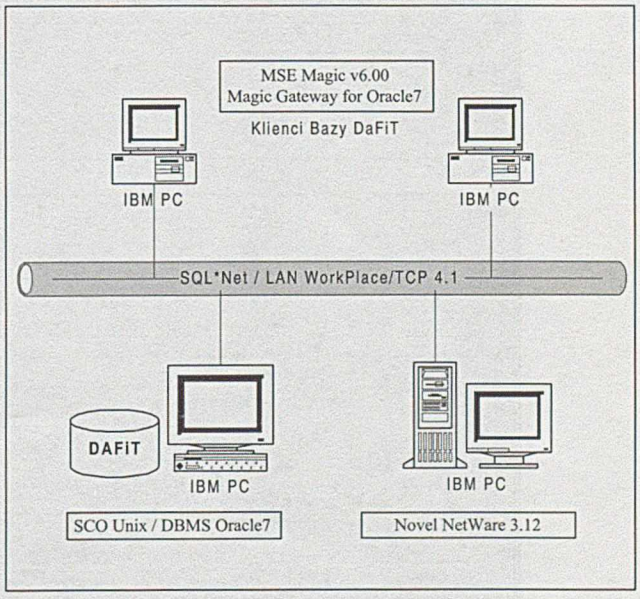
Pierwotnie stworzony system bazy DaFiT (wersja *„alfa"*) pracuje w heterogenicznym środowisku systemów operacyjnych SCO OpenServer v. 5.0, MS Windows i Novell NetWare. Całość oprogramowania użytkowego systemu wykonano przy użyciu generatora aplikacji Magic v.6 for Windows. Przy projektowaniu aplikacji szczególną uwagę zwrócono na zapewnienie pełnej przenośności oprogramowania pomiędzy różnymi systemami zarządzania bazami danych (*SQL* i pozostałymi).

Z uwagi na fakt, iż językiem, w jakim zapisane są informacje w bazie, jest język angielski oraz ze względu na zamiar udostępnienia bazy danych w sieci rozległej, całość oprogramowania aplikacyjnego wykonano w wersji angielskiej.

System w wersji *„alfa"* przeznaczony jest do pracy w sieci lokalnej. Obecnie przygotowywana wersja 1.0 systemu udostępnia dane w środowisku sieciowym (Internet).

## Konfiguracja eksperymentalna systemu bazy DaFiT

Podstawowym elementem testowego systemu bazy DaFiT w wersji *„alfa"* jest serwer bazy danych: komputer z procesorem Cyrix 5x86-120MHz wyposażony w 32MB pamięci RAM i pracujący pod systemem operacyjnym SCO OpenServer wersja 5.0. Na serwerze zainstalowano oprogramowanie systemu zarządzania bazą danych Oracle7. Serwer posiada połączenie z lokalną siecią Novell NetWare. Oprogramowanie użytkowe pozwalające na dostęp do bazy danych – aplikację generatora MSE Magic 6.00 uruchomiono na komputerze PC wyposażonym w procesor Intel Pentium 120MHz, 32MB pamięci RAM i system operacyjny Windows for Workgroups 3.11. Komunikacja pomiędzy serwerem bazy danych, a stacją roboczą odbywa się za pomocą serwera sieci Novell NetWare 3.12 z zainstalowanym pakietem LAN WorkPlace/TCP 4.1, z wykorzystaniem protokółu SQL*Net. Aplikacja Magic'a uzyskuje dostęp do zasobów bazy danych poprzez zastosowanie gateway'a do tej bazy (Magic Gateway for Oracle7) – rys. 1.




**Rys.1. Schemat konfiguracji eksperymentalnej systemu bazy DaFiT.**

## Sformułowanie zadań dla systemu bazodanowego

Z uwagi na charakter bazy danych DaFiT, dominującą operacją, wykorzystywaną w trakcie jej użytkowania, jest operacja wyszukiwania. Uzupełnianie i poszerzanie zbioru posiadanych informacji następuje zgodnie z rytmem pojawiania się nowych publikacji opisujących eksperymenty. Operacje modyfikacji i usuwania danych z bazy są stosunkowo rzadkie.

Celem użytkownika przeszukującego bazę danych jest z reguły dotarcie do informacji o wynikach badań przeprowadzonych na określonej substancji lub grupie substancji spełniających pewne warunki. Kryteria wyszukiwania, w zależności od potrzeby użytkownika, mogą dotyczyć także: źródła informacji (renoma, dostępność czasopisma), autorów publikacji, lat wydania, zakresów zmian parametrów eksperymentów. Zakres wyspecyfikowanych warunków, jakie powinny spełniać wyniki, jest dynamiczny i może ulegać zmianom w toku pracy (zawężanie, poszerzanie kręgu przeszukiwań). Wśród operacji wyszukiwania najczęściej pojawiają się wyszukiwania według: synonimów nazw substancji, wzoru chemicznego substancji, identyfikatora substancji *chemical abstract,* cech substancji, rodzaju badanej własności, nazwisk autorów publikacji, tytułów publikacji, lat wydania publikacji, zakresów parametrów badań. Ustalone przez użytkownika kryteria wyszukiwania mogą dotyczyć wszystkich wyżej wymienionych warunków.

Poza typowymi operacjami dodawania, modyfikacji, usuwania i wyszukiwania danych, system pozwala na wydrukowanie, interesujących użytkownika, informacji.

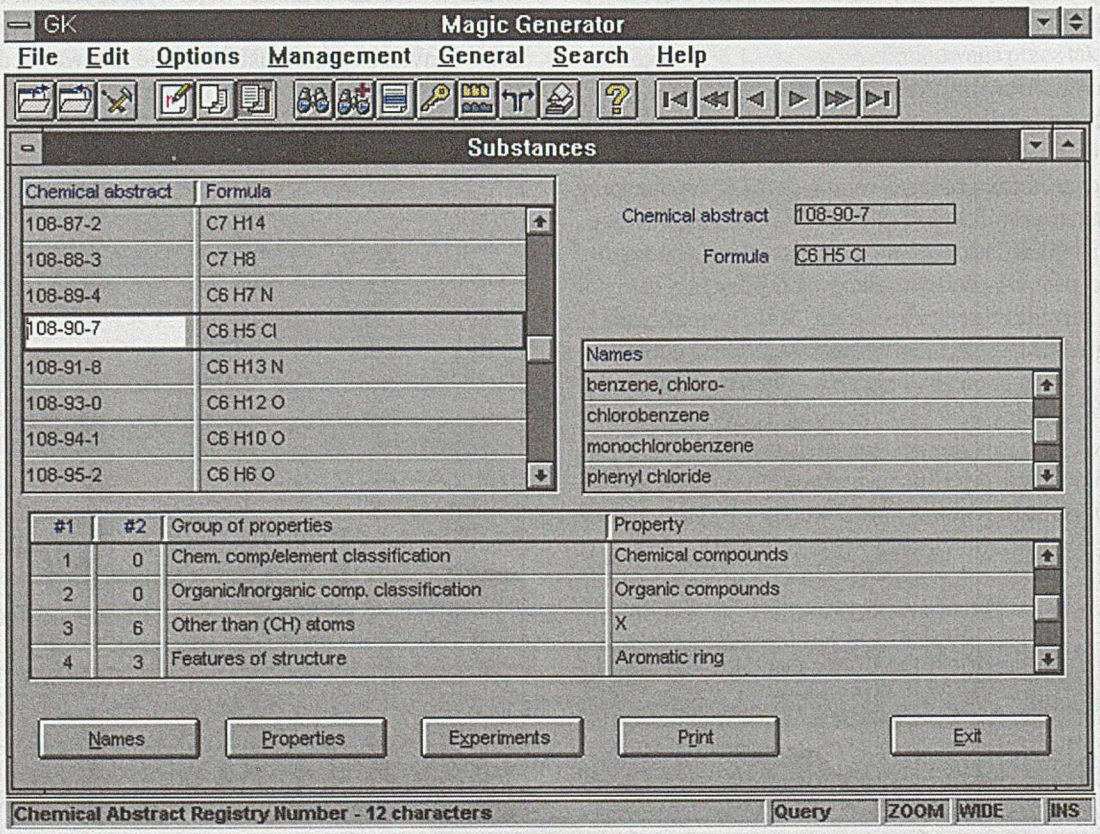
## Aplikacja użytkownika bazy DaFiT

W skład zbioru programów bazy DaFit w wersji „alfa" wchodzą: programy zarządzające danymi (wprowadzanie, modyfikacja, usuwanie), programy udostępniające dane oraz programy wyszukujące. Programy należące do dwóch pierwszych grup służą do wykonywania elementarnych operacji na zawartości bazy danych oraz dodatkowo posiadają możliwość ograniczania zakresu wyświetlanych rekordów, wyszukiwania rekordów, sortowania danych według określonego klucza. W celu realizacji tych operacji wykorzystują one możliwości jakie standardowo wbudowane są w każdą interakcyjną aplikację Magic'a *(Locate a Record, Locate Next, Range of Records, View by Key, Sort Records)*. Z uwagi na potrzebę ograniczenia dostępu do niektórych opcji (np. usuwania) programy te zintegrowano z systemem autoryzacji.

Szczególną grupę programów służących do wyszukiwania danych, zaprojektowano pod kątem uwzględnienia specyfiki przeszukiwania danych fizykochemicznych. Programy te pozwalają użytkownikowi na specyfikację złożonych warunków wyszukiwania odnoszących się do wielu tabel.

Z menu aplikacji, użytkownik ma bezpośredni dostęp do następujących programów:

Rys. 2. Główna formatka programu Substances



- *Browse ...* – proste programy służące do zarządzania danymi tabel bazy danych,
- *Substances* – program zarządzający substancjami chemicznymi (rys.2),
- *Experiments* – program zarządzający eksperymentami (rys.3),
- *Substance Search* – program służący do wyszukiwania substancji według podanych kryteriów,
- *Experiment Search* – program służący do wyszukiwania eksperymentów,
- *Experiment Search* (Restricted) – program służący do wyszukiwania eksperymentów z wykorzystaniem algorytmu zawężania obszaru przeszukiwań.

## Wybrane problemy projektowe

### *Wyszukiwanie wartości minimalnych i maksymalnych atrybutów w aplikacjach magic'owych*

W programie przedstawiającym wyniki eksperymentów, istnieje potrzeba wyświetlenia zakresu pierwszego parametru eksperymentu (zazwyczaj temperatur, w jakich dany eksperyment został przeprowadzony). Poniżej przedstawione są trzy rozwiązania tego problemu:

- użycie instrukcji SQL,
- sekwencyjne przeszukanie tabeli,
- wyszukanie z pomocą indeksu.

### *Użycie zadania typu SQL Command*

W przypadku SQL'owych baz danych, wyszukanie wartości minimalnej i maksymalnej najprościej zrealizować poprzez użycie zapytania:

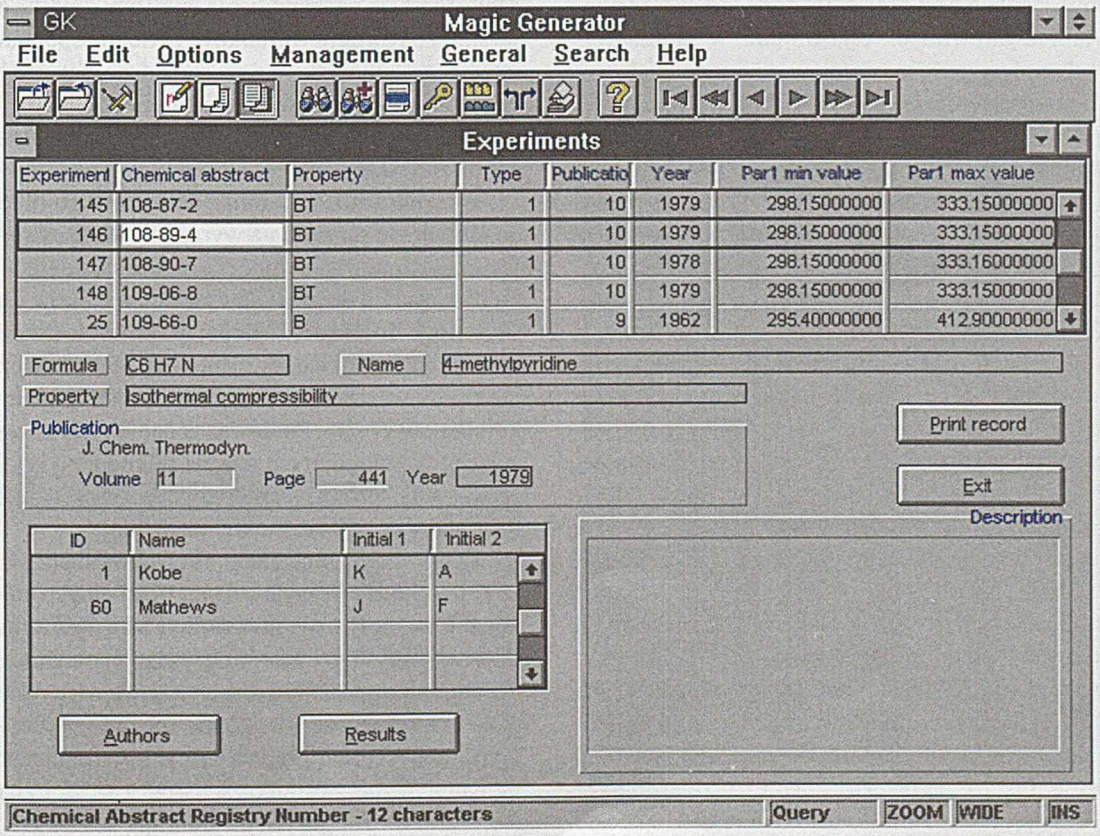
```
select min(par1_value), max(par1_value)
    from experiment_results;
```

Rozwiązanie to jest bardzo proste, ale nie jest przenośne (np. do Btrieve).

### *Sekwencyjne przeszukanie tabeli*

Metoda ta polega na przyjęciu za wartości minimalną i maksymalną wartości pierwszego pobranego rekordu danych i badaniu wzajemnych relacji między tymi wartościami a wartościami następnych rekordów. W przypadku napotkania wartości większej (mniejszej) od dotychczasowego maksimum (minimum), nowa wartość jest przyjmowana za nowe maksimum (minimum). Wadą tego rozwiązania jest konieczność przeglądnięcia całego zbioru rekordów tabeli i wynikająca stąd powolność tego procesu.

Rys. 3. Główna formatka programu Experiments



### *Wyszukanie z pomocą indeksu*

Dokonując indeksacji (załóżmy wzrastającej) na atrybucie, którego wartości ekstremalnych potrzebujemy, możemy sprowadzić cały proces wyszukiwania, do wykonania zaledwie dwóch operacji: instrukcji dołączenia (LINK) do zbioru z danymi i wyboru pierwszej napotkanej wartości. Dołączając się do zbioru po założonym indeksie z kierunku wzrastającego („od początku"; parametr *Dir: Asc*) dostajemy wartość minimalną atrybutu, dołączając się z kierunku malejącego („od końca"; parametr *Dir: Des*) otrzymujemy wartość maksymalną. Wykonanie operacji LINK w fazie Record–Main zadania głównego pozwala ponadto na umieszczenie wartości ekstremalnej atrybutu (np. zakresu temperatur eksperymentu) w wyświetlanej przezeń tabeli, gdyż pola plików dołączonych instrukcją LINK są dostępne w momencie tworzenia widoku danych, jeszcze przed wykonaniem faz związanych z przetwarzaniem rekordu.

Metoda wymaga istnienia dodatkowego indeksu. Z reguły indeks taki istnieje już w systemie, ponieważ często zachodzi potrzeba wyświetlenia danych w określonym porządku (np. wzrastająco, a nie w porządku wprowadzania), rozwiązanie to nie prowadzi więc w kierunku niepotrzebnego rozbudowywania aplikacji. Przedstawiona metoda jest bardzo szybka oraz uniwersalna dla Magic'a i z tych powodów została zastosowana w aplikacji.

### *Algorytmy wyszukiwania danych*

Jednym z głównych celów jaki ma realizować aplikacja jest umożliwienie użytkownikom łatwego wyszukiwania danych określonego typu. W czasie pracy z programami wyszukującymi użytkownik ma możliwość określenia wielu cech eksperymentów i substancji, których dane go interesują. Sposób w jaki użytkownik określa warunki w bardzo znaczący sposób wpływa na czas dotarcia do końcowych wyników (lub stwierdzenia, że wyników brak). W procesie wyszukiwania użytkownik zwykle postępuje według jednej z metod skrajnych: definiuje od razu wszystkie warunki lub deklaruje warunki po jednym, zawężając zakres uzyskiwanych wyników częściowych, lub też reprezentuje jakieś podejście pośrednie. Pierwsza z metod pozwala na najszybsze uzyskanie wyników, o ile takie wyniki istnieją. Porównując ilość kombinacji warunków z ilością informacji znajdującą się w bazie można stwierdzić, że dokładne trafienie w rekord danych jest mało prawdopodobne. Przy takim podejściu, bardzo duże znaczenie ma znajomość danych jakie są przechowywane w bazie – gdy wiadomo, jakie mniej więcej dane są dostępne, to dużo łatwiej je odnaleźć. Podejście bazujące na stopniowym definiowaniu warunków wyszukiwania jest godne polecenia przy małej ilości danych określonego typu. Specyfikując coraz to bardziej dokładne warunki, stopniowo zawęża się ilość danych, wśród których mogą znajdować się interesujące wyniki.

Jeżeli warunki narzucone na dane dotyczą pojedynczych wartości pól lub pewnych zakresów wartości, to do wyszukiwania należy stosować narzędzia dostarczone przez generator: opcje *Locate a Record, Locate Next, Range of Records, Sort Records, View by Key,* mechanizm *Smart Query*. Użycie tych mechanizmów jest stosunkowo proste, a czasy wyszukiwania są najmniejsze. W przypadku złożonych warunków narzuconych na dane należy opracować własne, specjalizowane programy wyszukujące. Algorytm działania programu wyszukującego można zasadniczo oprzeć o dwie metody :

- wykorzystanie wyrażenia *Range Exp,*
- wykorzystanie dodatkowego pliku (tabeli).

Wyrażenie *Range Exp* dialogu *Objects / Task Control* służy do zdefiniowania warunku logicznego jaki muszą spełniać pola danych rekordu, aby mógł on być wyświetlony na formatce ekranowej. Dla rekordów spełniających wszystkie, podane przez użytkownika, warunki wyrażenie to powinno przyjąć wartość TRUE.

Przy definiowaniu wyrażenia *Range Exp* należy wziąć pod uwagę, że jest ono obliczane jeszcze przed fazą Record Prefix i z tego powodu w wyrażeniu mogą brać udział tylko zmienne udostępnione instrukcją SELECT i LINK oraz wartość parametru *Ret* zwracana przez operację LINK. Wyrażenie *Range Exp* deklaruje się je zwykle na poziomie zadania, które zajmuje się wyświetlaniem danych.

Główną zaletą stosowania wyrażenia *Range Exp* jest uniknięcie konieczności stosowania dodatkowego, pomocniczego pliku. Wadą wynikającą z użycia tego rozwiązania jest znaczne spowolnienie procesu przeszukiwania. Zastosowanie wyrażenia *Range Exp* powoduje, że generator będzie wykonywał liniowe przeszukiwanie pliku głównego i zbiorów dołączonych. Przy znacznej ilości danych czas wykonania procesu wyszukiwania może sięgać godzin.

Z wyrażenia *Range Exp* korzystają programy *Experiment Search* i *Substance Search*. Programy te przeprowadzają operacje wyszukiwania na całym zbiorze danych: po wyspecyfikowaniu warunków, jakie ma spełniać eksperyment, przeszukiwane są wszystkie dane zawarte w bazie, po ewentualnej zmianie warunków (zawężeniu, rozszerzeniu), w kolejnym procesie przeszukiwania, ponownie są przeglądane wszystkie dane.

Wyszukiwanie z pomocą dodatkowego pliku pozwala na uniknięcie stosowania wyrażenia *Range Exp*. Algorytm wykorzystujący tę metodę posługuje się pomocniczym plikiem służącym do przechowywania wyników wyszukiwań częściowych. Każde, następujące po sobie uszczegółowienie warunków nakładanych na dane, powoduje przeglądnięcie zbioru wcześniej wybranych danych i jego ewentualne, kolejne zawężenie. Ponieważ zadanie wyznaczające nowy, zawężony zbiór danych stosuje tylko jedno kryterium wyszukiwania, nie musi ono korzystać z wyrażenia *Range Exp* – działa więc ono stosunkowo szybko.

W pierwszym kroku algorytmu w skład przeszukiwanego zbioru wchodzą wszystkie dane. W kolejnych krokach użytkownik ma możliwość stwierdzenia jaką ilość informacji uzyskał i z której strony należy dążyć do jej ograniczenia.

Ograniczenie zbioru przetwarzanych danych pozwala na stosunkowo szybkie tworzenie specyficznych, kontekstowych list wyboru uwzględniających warunki narzucone już na dane. Wybór warunku spośród wartości znajdujących się w zbiorze roboczym gwarantuje, że zawsze uzyskamy

jakąś informację, choćby tę, na podstawie której warunek został ustalony. Podobne rozwiązanie, związane z listami wyboru, można stosować i w metodzie przeszukującej cały zbiór danych, ale jest ono, w tamtym przypadku, bardzo czasochłonne.

Program rozpoczyna pracę z pustym plikiem rozwiązań częściowych (dokładnie: brak jest w pliku rekordów sygnowanych unikalnym identyfikatorem zadania). Po specyfikacji pierwszego z warunków następuje przeszukanie zawartości bazy danych, a dane które spełniają wymagane kryterium zostają przepisane do pliku rozwiązań częściowych. W przypadku dalszej specyfikacji warunków, przetwarzanie ogranicza się tylko do tego zbioru.

W celu oszacowania wydajności systemu DaFiT pracującego na DBMS Oracle7, sprawdzono czasy wykonania niektórych operacji systemu. Używana podczas testu baza danych zawierała rzeczywiste informacje o 2449 substancjach, 7036 synonimach nazw substancji, 12546 symbolicznych opisach właściwości substancji, 21208 eksperymentach, 164743 wynikach eksperymentów, 3778 autorach i 2510 tytułach publikacji. Po przeprowadzeniu szeregu badań stwierdzono, że:

- standardowe operacje oferowane przez system: lokalizacja danych, ograniczanie widoczności danych, sortowanie i wyświetlanie wyników według określonego klucza są wykonywane bardzo szybko (wymagają około 1s.), o ile wykorzystują istniejące w bazie indeksy. Wykonanie tych samych operacji na danych, na których nie założono indeksu, może już trwać dziesiątki minut (np. posortowanie danych według złożonego klucza),
- stosunkowo długo trwa proces uruchamiania poszczególnych programów aplikacji. Od wyboru opcji z menu (wywołania programu), do wypełnienia tabel formatek i przekazania sterowania użytkownikowi mija czas od kilku do kilkunastu sekund,
- najbardziej czasochłonne są procesy wyszukiwania przeprowadzane według złożonych kryteriów. Wśród grupy programów wyszukujących najlepiej prezentują się programy nie wykorzystujące przy wyświetlaniu danych wyrażenia *Range Exp*, co pozwala generatorowi na sprawniejszą pracę z użyciem indeksów. Należący do nich program *Experiment Search (Restricted)*, działa stosunkowo szybko, ale zawiera on ekstremalną, jak na aplikację magic'ową, liczbę szesnastu specjalizowanych podzadań wyszukujących.

## Kontrola dostępu do danych

Z uwagi na bardzo duże gospodarcze znaczenie danych przechowanych w bazie DaFiT należało zapewnić odpowiedni poziom ich bezpieczeństwa.

Magic zawiera w sobie elastyczny system autoryzacji pozwalający na szczegółową kontrolę dostępu użytkowników do obiektów aplikacji i samych danych. System, najogólniej mówiąc, oparty jest na przydziale praw. Prawa w Magic'u mogą być nadawane indywidualnym użytkownikom oraz grupom.

### *Prawa, grupy i użytkownicy zdefiniowani w systemie bazy DaFiT*

Lista praw zdefiniowanych w systemie DaFiT:

- **Super Right** – superprawo – klucz pełnych praw dostępu do wszystkich obiektów i możliwości aplikacji. Posiadacz klucza ma nieskrępowany dostęp do każdego z elementów aplikacji i nie potrzebuje mieć przydzielonych, żadnych innych jednostkowych praw. Prawo to zawiera w sobie także uprawnienia związane z prawami *Application Access* i *Public Rights Access*;
- **Application Access** – prawo dostępu do aplikacji, tylko użytkownicy posiadający klucz mogą otworzyć aplikację (aplikację może otworzyć także posiadacz klucza *Super Right*);
- **Public Rights Access** – prawo dostępu do praw publicznych, istnienie tego klucza zwiększa bezpieczeństwo systemu;
- **Modify CTL Properties** – prawo pozwalające na modyfikowanie zawartości pól dialogu *CTL Properties;*
- **Adm Program** – prawo wykonywania programów administratorskich;
- **Modify Delete Data** – prawo do modyfikacji i usuwania danych;
- **Create Data** – prawo do tworzenia nowych rekordów danych.

W systemie przyjęto zasadę nadawania praw grupom, a nie poszczególnym użytkownikom. Utworzono trzy grupy użytkowników:

- ***PhysChemAdmGroup*** – grupa administratorów bazy danych, z przypisanym prawami: *Application Access, Modify CTL Properties, Adm Program, Modify Delete Data, Create Data*. Użytkownicy należący do tej grupy sprawują kontrolę nad całością danych zawartych w bazie i mogą wykonywać na nich wszelkie operacje;
- ***PhysChemCreateGroup*** – grupa użytkowników bazy, z prawami *Application Access i Create Data*. Członkowie grupy posiadają prawo do wstawiania nowych danych do bazy i ograniczone prawo do modyfikacji danych (wymagane w zadaniach *Experiments* i *Substances*). Nie mogą oni natomiast usuwać już istniejących informacji;
- ***PhysChemUsersGroup*** – grupa użytkowników bazy, tylko z prawem *Application Access*. Grupa zwykłych użytkowników upoważniona tylko do przeglądania zawartości bazy danych (z możliwością wyszukiwania informacji).

### *Autoryzacja dostępu do DBMS Oracle7 z poziomu Magic'a*

W systemie DaFiT, każde z narzędzi: DBMS Oracle7 i Magic ma swój własny system autoryzacji. Istnienie dwóch niezależnych systemów autoryzacji wymusza dwufazowy proces logowania się do bazy danych. W pierwszej fazie użytkownik loguje się do systemu Magic, następnie poprzez Magic'a do DBMS Oracle7. Przy logowaniu się do bazy danych Magic wykorzystuje informacje zawarte w polach *User Name* i *User Password* dialogu *Settings / Databases / Database Properties*. Pola te zawierają nazwę i hasło użytkownika istniejącego w DBMS'ie, na którym pracuje Magic. W celu usprawnienia dostępu do bazy danych można ustalić zawartość tych pól. W takim wypadku, przy otwieraniu aplikacji, użytkownik zwolniony jest z wypełniania ich zawartości, a przez to i oddzielony od spraw związanych z systemem autoryzacji DBMS'u.

### *Przenośność aplikacji do innych systemów zarządzania bazami danych*

Jedną z największych zalet aplikacji generatora Magic jest ich niezależność od platformy bazodanowej. Warunkiem zachowania tej niezależności jest rezygnacja ze stosowania specyficznych mechanizmów danego systemu. W szczególności chodzi tu także o nieużywanie instrukcji *SQL*.

Aplikacja bazy DaFiT została opracowana z wykorzystaniem generatora Magic i bazy danych DBMS Oracle7. Przeniesienie aplikacji do innego DBMS'u (np. do Btrieve) nie wymaga zmian w tekście programu aplikacji. Generalnie, programista musi tylko określić nowe położenie plików danych (w polach dialogu *File Dictionary: DB File, DBMS, Database*).

Przy pisaniu aplikacji **dedykowanej** tylko na DBMS Oracle7, można wykorzystać możliwość wykonywania, z poziomu Magic'a, instrukcji języka *SQL*. Bezpośrednie wyko- rzystanie mechanizmów DBMS'u pozwala na najszybsze wykonanie kierowanych doń instrukcji – Magic uczestniczy w tym procesie tylko w minimalnym stopniu, a wybór najefektywniejszego sposobu wykonania instrukcji (optymalizacja) pozostaje w gestii systemu zarządzania. Szczególnie wyrazistym przykładem takiej sytuacji może być zadanie modyfikacji wartości atrybutu tabeli. Korzystając z zadania typu *SQL Command* wykonujemy je pojedynczą instrukcją UPDATE, wykorzystując standardowe zadanie Magic'a należy przeglądnąć część lub cały plik danych (w celu pobierania jego kolejnych rekordów).

W przypadku aplikacji DaFiT, bardzo korzystne byłoby użycie instrukcji *SQL* w programach wyszukujących. Pozwoliłoby to na zastąpienie złożonych podzadań wyszukujących zagnieżdżoną instrukcją SELECT lub wywołaniem tzw. procedury składowanej (*stored procedure*) – przechowywanej, już w stanie skompilowanym, na samym serwerze bazy danych.

❖ ❖ ❖

Opracowany system bazy DaFiT w wersji *„alfa"* został ograniczony do zarządzania danymi o substancjach czystych. Realizuje on wszystkie podstawowe operacje na bazie danych, takie jak: przeglądanie zawartości bazy, wprowadzanie, modyfikacja i usuwanie danych. System zapewnia możliwość szybkiego wyszukania interesujących informacji przez wykorzystanie standardowych, wbudowanych weń mechanizmów lub skorzystanie ze specjalnych programów wyszukujących.

Działanie systemu bazy danych można przyspieszyć przez: uruchomienie go na sprzęcie o wysokiej mocy obliczeniowej (docelową platformą, na której będzie umieszczona baza danych DaFiT, będą komputery SUN) lub też napisanie specjalizowanej aplikacji wykorzystującej pewne specyficzne cechy DBMS Oracle7. W celu zapewnienia efektywnego i sprawnego współdziałania generatora Magic z bazą danych należy także odpowiednio skonfigurować sam serwer bazy danych.

***Baza Danych Fizykochemicznych i Termofizycznych** powstała w wyniku współpracy Zakładu Karbochemii PAN w Gliwicach, Instytutu Chemii Fizycznej PAN w Warszawie, Thermodynamic Research Center of the Texas A&M University (TRC w USA) oraz CODATA Task Group on Phase Equilibrium Data i World Data Depository on Thermophysical Property Data.*

**Marcin Gorawski** jest pracownikiem Pracowni Podstaw i Zastosowań Informatyki Zakładu Karbochemii PAN w Gliwicach.

**Jacek Frączek** jest pracownikiem Instytutu Informatyki Politechniki Śląskiej w Gliwicach.

## Potrzebna pomoc

Fundacja Pomocy Matematykom i Informatykom Niesprawnym Ruchowo, która po raz pierwszy w siedmioletniej historii, uzyskała w budynku PAN przy ul. Twardej 51/55 w Warszawie lokal na Komputerowe Laboratorium Inwalidów Ruchu (**KLABIR - tel. 6978 - 784, tel./fax 6978 -786, e-mail: kmark@pol.pl**), otrzymała w ostatnim okresie od firm komputerowych kilka cennych darów.

Firma OKI wyposażyła lokal w faks wartości kilkunastu tysięcy zł. Od firmy KARMA Polska Fundacja otrzymała dar wartości ponad 24 tys. zł w postaci płyt głównych, dysków, CD-ROM-ów i pamięci. TCH Components przekazała płyty główne i obudowy wartości ponad 20 tys. zł. Części te pozwolą na modernizację sprzętu znajdującego się w Komputerowym Laboratorium Inwalidów Ruchu i w domach podopiecznych Fundacji.

Obecnie Fundacja przygotowuje się do uruchomienia I Węzła Internetu dla Niepełnosprawnych, który umożliwi inwalidom ruchu pracę i uzupełnianie edukacji w domu oraz kontakt ze światem. Węzeł będzie uruchomiony na serwerze ofiarowanym przez firmę IBM i oprogramowaniu przekazanym Fundacji przez Microsoft Polska. Router jest darem firm CISCO i SOLIDEX. Akcję tę wspiera TP S.A. i Fundacja Stefana Batorego.

FPMiINR stara się na siebie zarabiać. Podopieczni prowadzą kursy i wykonują zlecenia informatyczne, co przynosi Fundacji skromny zysk. Korzysta też z pracy wielu wolontariuszy oraz otrzymuje dary od indywidualnych ofiarodawców, firm i sporadyczne dotacje. Niestety, pomocy omówił Państwowy Fundusz Reahabilitacji Osób Niepełnosprawnych, który nie docenia tej formy nowocześnie pojmowanej rehabiltacji zawodowej inwalidów, jaką stwarza informatyka, a szczególnie Internet.

Jak poinformowała nas Przewodniczaca Zarzadu Fundacji - prof. Elżbieta Pleszczyńska brak stałych funduszy na elementarne wydatki sprawia, ze Fundacja w tej chwili wegetuje na granicy przetrwania. Uniemożliwia to systematyczną pracę, marnotrawi wieloletnie wysiłki i szanse. Namiastkę stabilizacji finansowej zapewnić może tylko rodzaj stałego patronatu, w ramach którego indywidualni ofioarodawcy, firmy i instytucje zadeklarowałyby choćby najskromniejsze regularne wsparcie miesięczne lub roczne na rzecz Fundacji. Gorąco więc prosimy czytelników o wsparcie. Można kontaktować się bezpośrednio z Fundacją (adres i telefony podano wyżej) lub wpłacać dary na konto: **BGK I/0 Warszawa 11301017-10802047-27005-1.**

Więcej informacji o FPMiINR można znaleźć pod adresem Internetowym: **http://www.idn.pol.pl.**

SYSTEMY CZASU RZECZYWISTEGO

REDAGUJE ZESPÓŁ: dr inż. Zbigniew FRYŹLEWICZ (Politechnika Wrocławska), prof. dr hab. inż. Janusz GÓRSKI (Instytut Technik Telekomunikacyjnych i Informatycznych w Poznaniu), prof. dr hab. inż. Zbigniew HUZAR (Politechnika Wrocławska), prof. dr hab. inż. Tomasz SZMUC (Akademia Górniczo-Hutnicza), dr inż. Kazimierz WORWA (Wojskowa Akademia Techniczna), mgr inż. Zdzisław ŻURAKOWSKI (Instytut Automatyki Systemów Energetycznych we Wrocławiu) - przewodniczący zespołu.

ROK IV • CZERWIEC • 1997 • 29

Metody Analizy Bezpieczeństwa Systemów Komputerowych

# Wprowadzenie do cyklu

Janusz Górski

Cykl artykułów, który rozpoczynamy w tym numerze Wkładki *Systemy Czasu Rzeczywistego* jest poświęcony metodom analizy bezpieczeństwa systemów komputerowych. Komputery sterują samolotami, samochodami, pociągami, sprzętem wojskowym, urządzeniami medycznymi, centralami telefonicznymi, są współcześnie obecne w nieomal każdej sferze naszego życia zawodowego i prywatnego. Wiele z tych obszarów zastosowań wiąże się ze znacznym ryzykiem. Pojawia się więc problem bezpiecznego stosowania technologii informatycznej. Znane są wypadki, za które pełną odpowiedzialność ponosi oprogramowanie, np. system medyczny Therac-25, który w latach 1986-88 spowodował śmierć kilku osób czy ostatnio głośny przypadek rakiety Ariane 5. Bezpieczeństwo jest rozumiane jako atrybut całego systemu. Oznacza, że użytkowanie systemu nie doprowadzi do wypadków. System stanowi tu zespół urządzeń, najczęściej wykonanych w różnych technologiach, współpracujących w celu spełnienia określonej misji. Jego częścią może być system komputerowy (złożony z urządzeń komputerowych i zawartego w nich oprogramowania), który zwykle wypełnia funkcje pomiarowo-sterujące. Pozostała część systemu to system sterowany. System sterowany może być na przykład systemem mechanicznym, chemicznym, hydraulicznym lub nuklearnym.

## Bezpieczeństwo

Bezpieczeństwo jest atrybutem całego systemu. Zawarty w nim system komputerowy jako taki nie może w żaden sposób wywołać zdarzenia niebezpiecznego (wyjątkiem może być porażenie prądem elektrycznym lub upadek monitora na podłogę). Natomiast sygnały generowane przez komputer mogą służyć do sterowania innymi urządzeniami, a pewne sekwencje takich sygnałów mogą spowodować wypadek. Taki system jako całość może być niebezpieczny dla swojego środowiska. W tym kontekście można powiedzieć, że system komputerowy ma wpływ na bezpieczeństwo całego systemu, jest więc *podsystemem związanym z bezpieczeństwem*. W skrócie mówi się często o bezpieczeństwie systemu komputerowego jako takiego. Jest to powszechnie przyjęte uproszczenie, jednak nie można zapominać, że bezpieczeństwo ma sens tylko jako atrybut całego systemu.

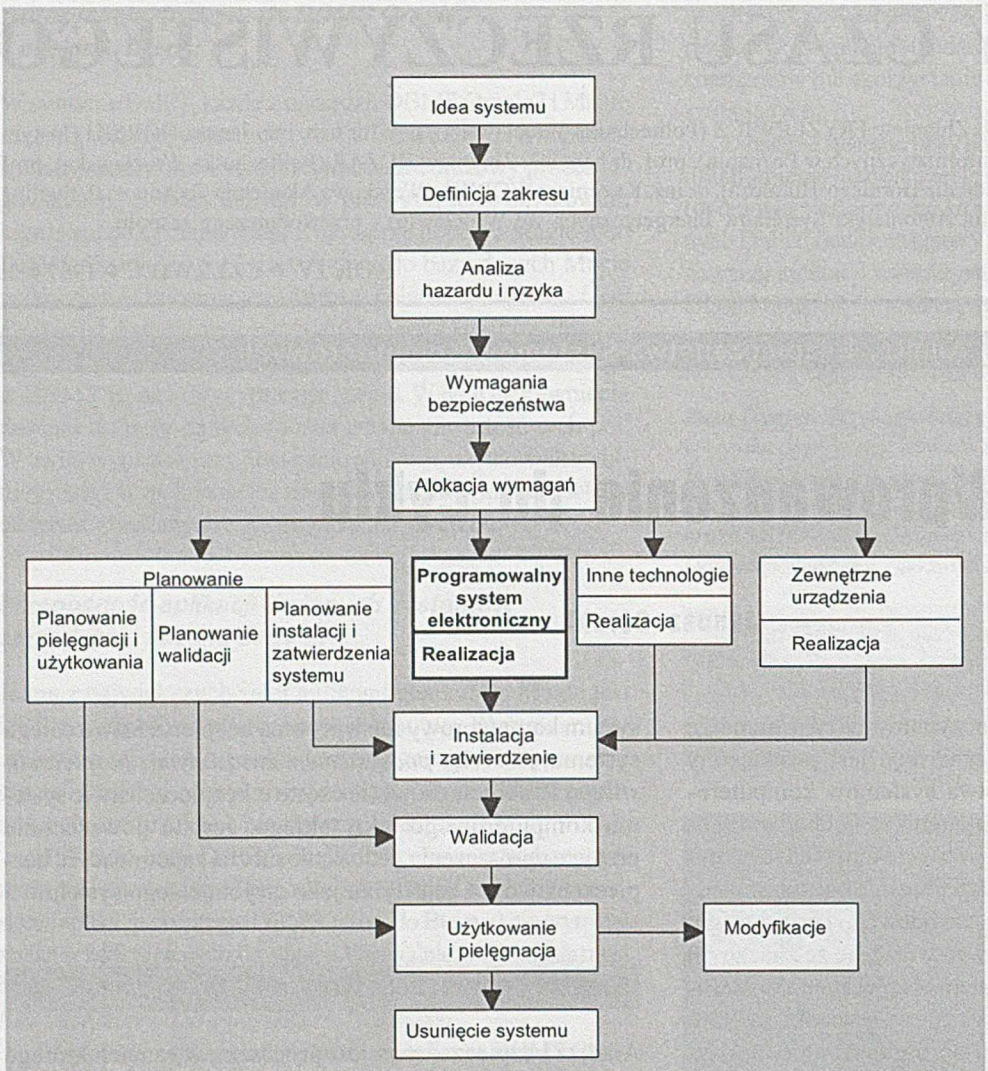
## Analiza bezpieczeństwa

Analiza bezpieczeństwa jest procesem, w ramach którego oceniany jest poziom ryzyka związanego z systemem i identyfikowane są mechanizmy występowania wypadków. Zdarzenia prowadzące w sposób bezpośredni do wypadku nazywane są stanami hazardowymi lub hazardami. Analiza bezpieczeństwa jest częścią cyklu życia bezpieczeństwa. Na Rys.1. przedstawiono model cyklu życia bezpieczeństwa, według standardu IEC 1508 (*IEC - International Electrotechnical Commission*). Standard ten, przewidywany do wprowadzenia w roku bieżącym, obejmuje zagadnienie stosowania komputerów w systemach związanych z bezpieczeństwem. Należy oczekiwać, że po wprowadzeniu stanie się on wzorcem dla opracowywania bardziej szczegółowych standardów branżowych.

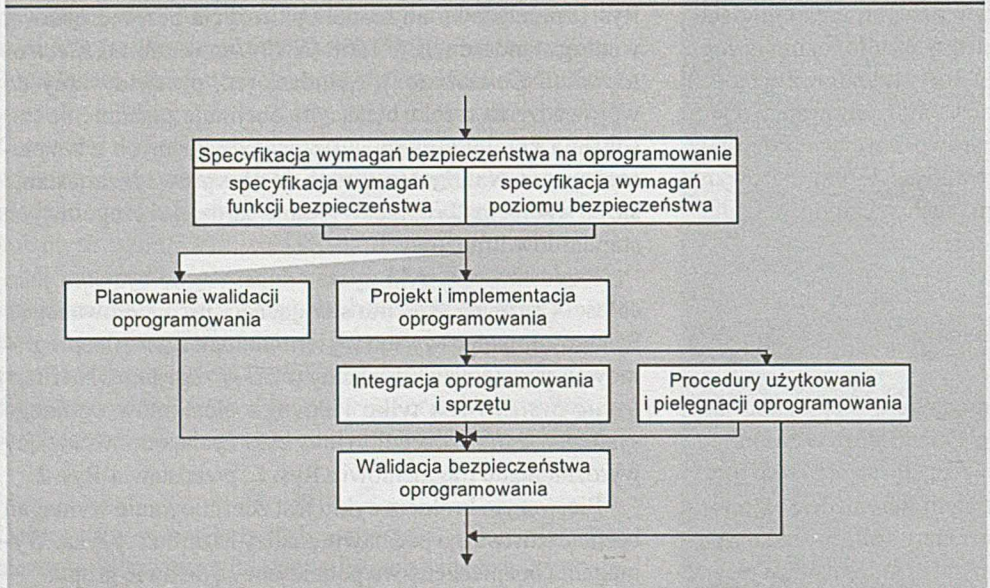
Przedstawiony cykl życia odnosi się do systemu jako całości (zarówno systemu sterującego, jak i sterowanego). System komputerowy, lub wg terminologii IEC1508, programowalny system elektroniczny (*PES - Programmable Electronic System*) jest tylko jednym z elementów ogólnego systemu. Jego uszczegółowiony cykl życia, odpowiadający wyróżnionemu fragmentowi z Rys. 1., przedstawia Rys. 2.

Pierwszym krokiem cyklu jest zdefiniowanie wymagań bezpieczeństwa, na podstawie analizy hazardu i ryzyka. Wymagania bezpieczeństwa podzielone są na dwie grupy:

- **wymagania funkcji bezpieczeństwa,**
- **wymagania poziomu bezpieczeństwa.**




Rys. 1. Cykl życia bezpieczeństwa systemu.




Rys. 2. Cykl życia bezpieczeństwa programowalnego systemu elektronicznego.

Pierwsza grupa definiuje bezpieczne zachowania systemu. Druga grupa wymagań określa pożądany poziom pewności, że dana funkcja bezpieczeństwa będzie poprawnie zrealizowana. Wybór poziomu bezpieczeństwa pociąga za sobą konieczność zastosowania określonych technik wytworzenia systemu. Poziom bezpieczeństwa jest ustalany na podstawie analizy ryzyka, dla każdej funkcji oddzielnie. Im większe ryzyko związane z daną funkcją, tym wyższy jest wymagany poziom bezpieczeństwa. Standard IEC1508 wyróżnia cztery poziomy bezpieczeństwa, od 1 do 4. Wymagania dotyczące poziomu 4 są najwyższe i obejmują między innymi zastosowanie metod formalnych podczas analizy, projektowania i weryfikacji oprogramowania. Poziom 1 jest najniższym poziomem omawianym przez standard. Jednym z wymagań wyjściowych dla osiągnięcia poziomu 1 (i wyższych) jest zastosowanie norm ISO 9001 w procesie wytwarzania systemu.

Analiza bezpieczeństwa jest procesem mającym na celu określenie stopnia bezpieczeństwa systemu i obejmuje:

- analizę hazardu – obejmującą identyfikację zdarzeń niebezpiecznych i ich przyczyn,
- analizę ryzyka – obejmującą identyfikację częstości i skutków zdarzeń niebezpiecznych.

Analiza bezpieczeństwa jest procesem ciągłym, wykonywanym w ramach kolejnych etapów cyklu życia systemu:

- podczas tworzenia systemu, w celu zidentyfikowania zagrożeń i dobrania właściwych środków projektowania i realizacji systemu,

- podczas eksploatacji, w celu sprawdzenia i/lub poprawienia stopnia bezpieczeństwa systemu (obejmuje to także analizy wykonywane po wypadkach),
- podczas oceny systemu przez niezależną instytucję certyfikującą, dla sprawdzenia stopnia bezpieczeństwa i zaakceptowania (lub odrzucenia) systemu do użytkowania.

Cele analizy bezpieczeństwa są osiągane przez zastosowanie różnych technik. Nie istnieje obecnie jedna, uniwersalna i „najlepsza" technika analizy bezpieczeństwa. W praktyce, argument na rzecz bezpieczeństwa systemu ma złożoną strukturę, w ramach której uwzględniane są rezultaty wielu analiz, przeprowadzonych różnymi technikami. Niektóre z tych technik uzupełniają się wzajemnie. Ale nawet w wypadku gdy zakresy różnych technik pokrywają się, ich łączne zastosowanie może być celowe, gdyż jeżeli potwierdzą wzajemnie swoje rezultaty, można mieć większe zaufanie do przeprowadzonych analiz.

## Cel cyklu

Celem niniejszego cyklu artykułów prezentowanego Czytelnikom jest zaprezentowanie wybranych technik analizy bezpieczeństwa. Wyboru dokonano pod kątem prezentacji tych technik, które mają istotne znaczenie przy analizie związanych z bezpieczeństwem systemów komputerowych. Są to techniki tradycyjne, stosowane wcześniej do systemów konwencjonalnych, dla których zaproponowano już rozszerzenia i modyfikacje umożliwiające ich stosowanie również w odniesieniu do systemów komputerowych (takie jak Analiza Drzew Błędów, HAZOP czy FMEA), techniki specjalnie stworzone dla analizy systemów komputerowych, takie jak obiektowa analiza bezpieczeństwa czy wreszcie inspekcje - ogólna metoda analizy jakości dokumentacji.

Prezentacje dotyczą tych technik, które są używane i rozwijane w ramach Centrum Inżynierii Oprogramowania (CIO), instytucji ukierunkowanej na poprawę jakości procesów programistycznych w zastosowaniach przemysłowych i transfer technologii w tym zakresie. Współautorami artykułów są Koledzy z Centrum oraz specjaliści z innych instytucji, współpracujący z nami w zakresie zagadnień związanych z bezpieczeństwem i jakością systemów komputerowych.

## Struktura cyklu

W ramach cyklu przedstawione zostaną nastepujące artykuły:

1. *Łącznie z niniejszym wprowadzeniem ukaże się artykuł Andrzeja Wardzińskiego z CIO oraz Marko Čepina z Instytutu Jozefa Stefana z Lubljany, prezentujący metodę analizy drzew błędów. Autorzy stosowali i rozwijali tę metodę w odniesieniu do systemu zabezpieczeń elektrowni jądrowej Krsko w Słowenii.*

2. *Marek Jurkowlaniec i Bartosz Nowicki z CIO oraz Felix Redmill z Redmill Consultancy w Londynie przedstawią w kolejnym artykule cyklu metodę analizy hazardów systemowych HAZOP. Jest to metoda zapoczątkowana w przemyśle chemicznym, szybko zyskująca popularność w innych obszarach zastosowań. Felix Redmill był jednym z autorów standardu związanego z tą metodą.*

3. *Następnym w cyklu będzie artykuł Tadeusza Cichockiego z ADTranz Zwus Signal w Katowicach na temat metody FMEA. ADTranz Zwus Signal współpracuje z CIO w zakresie bezpieczeństwa komputerowych systemów sygnalizacji kolejowej. FMEA jest jedną z metod analizy tradycyjnie stosowanej w odniesieniu do takich systemów.*

4. *Następnie Bartosz Nowicki z CIO oraz Steve Riddle z Newcastle University przedstawią podstawy obiektowego podejścia do analizy bezpieczeństwa systemów komputerowych. Artykuł będzie zawierał niektóre wyniki uzyskane w tym zakresie w trakcie wspólnie realizowanego projetu badawczego w programie Copernicus.*

5. *Jako ostatni przedstawiony zostanie artykuł Marka Jurkowlańca z CIO oraz Bjorna Runge z Dansk Data Elektronik A/S w Danii, którzy przedstawią metodę inspekcji Fagana, jedną z najskuteczniejszych metod analizy jakości dokumentacji, w tym również oprogramowania. Bjorn Runge jest znanym ekspertem w zakresie metod inspekcji. Metody te nie są bezpośrednio związane z analizą bezpieczeństwa. Są to jednak metody o znacznym potencjale związanym z polepszeniem jakości i wczesnym wykrywaniem defektów dokumentów. Z tego względu uznałem, że warto zwrócić na nie uwagę Czytelników cyklu, zainteresowanych bezpieczeństwem (a więc również jakością) zastosowań informatyki.*

**prof. Janusz Górski** jest pracownikiem Centrum Inżynierii Oprogramowania, ITTI Zakład Zastosowań Informatyki, Politechnika Gdańska

# Analiza drzew błędów

**Andrzej Wardziński, Marko Čepin**

Analiza drzew błędów (*Fault Tree Analysis*, FTA) jest jedną z podstawowych, szeroko stosowanych metod analizy bezpieczeństwa. Jej celem jest identyfikacja przyczyn niebezpiecznego stanu systemu oraz ocena możliwości jego wystąpienia. Technika ta jest od lat z powodzeniem stosowana dla systemów tradycyjnych (niekomputerowych) oraz komputerowych. Jest szczególnie popularna w przemyśle lotniczym, elektronicznym oraz nuklearnym.

Metodę analizy drzew błędów po raz pierwszy zastosował H. A. Watson w Bell Telephone Laboratories (USA) w roku 1961 w celu oceny systemu kontroli odpalania pocisku Minuteman. Badano możliwość nieautoryzowanego odpalenia pocisku. Metoda ta wykorzystywała logikę boolowską do analizy przyczyn awarii. Istotny wkład w rozwój metody mieli inżynierowie i matematycy firmy lotniczej Boeing. Dodali oni możliwość analizy probabilistycznej drzew błędów.

Celem metody FTA jest analiza przyczyn niebezpiecznego zachowania systemu. Ryzyko (ang. *hazard*) rozumiane jest jako fizyczna sytuacja potencjalnie prowadząca do dużych strat materialnych, zagrożenia zdrowia lub życia człowieka [1]. Może to być zdarzenie lub stan systemu lub jego elementu prowadzące w sposób bezpośredni do wypadku (bezpośrednia przyczyna wypadku). Na podstawie analizy drzew błędów można powiedzieć, jakie są przyczyny ryzyka, jakie scenariusze zdarzeń do niego prowadzą oraz jak często on występuje.

Wyróżniamy dwa rodzaje analizy drzew błędów: jakościową oraz ilościową. Podejście jakościowe jest oparte o wnioskowanie matematyczne i pozwala wypowiadać się o zależnościach przyczynowo-skutkowych. Na tej podstawie możemy powiedzieć, jakie są przyczyny niebezpiecznego stanu i jakie muszą pomiędzy nimi zajść relacje, aby spowodować awarię systemu. Podejście ilościowe pozwala określić prawdopodobieństwo tej awarii oraz jej przyczyn. Znajomość najbardziej prawdopodobnych przyczyn pozwala w efektywny sposób zabezpieczyć system (to znaczy zmniejszyć możliwość osiągania niebezpiecznego stanu).

Porównując FTA z innymi metodami analizy bezpieczeństwa, należy mieć na uwadze dwie cechy metody. Po pierwsze, nie służy ona do identyfikacji stanów niebezpiecznych ale gdy są znane pozwala zidentyfikować ich przyczyny. Kolejną szczególną cechą jest kolejność analizy. Punktem wyjściowym jest niebezpieczny stan całego systemu. Podczas konstrukcji drzewa określane są coraz bardziej szczegółowe przyczyny tego stanu. Granicą jest osiągnięcie poziomu awarii pojedynczych elementów systemu. Analiza wiedzie więc od systemowego stanu niebezpiecznego do pojedynczych elementów systemu.

Metoda FTA jest zdefiniowana przez standard Międzynarodowej Komisji Elektrotechnicznej IEC 1025. Standard ten opisuje cele i kroki metody (nie zawiera jednak szczegółowej procedury postępowania). Stosowanie metody jest wymagane przez standardy dotyczące bezpieczeństwa, na przykład w przemyśle nuklearnym, lotniczym, kolejowym, morskim. Obecnie powstaje standard IEC 1508 dotyczący bezpieczeństwa zastosowań systemów komputerowych [1]. Jego wydanie jest oczekiwane w roku 1997. Po wejściu w życie stanie się on podstawowym standardem normującym zastosowania komputerów w systemach związanych z bezpieczeństwem. Metoda FTA jest jedną z metod, których stosowanie jest wymagane przez ten standard.

## Cele i kroki metody

Warunkiem zastosowania metody FTA jest znajomość konstrukcji systemu. Metoda ta jest często wykorzystywana do analizy wstępnej projektu systemu – taki rodzaj analizy nazywany jest wstępną analizą ryzyka (*preliminary hazard analysis*). Jej celem jest analiza stanów niebezpiecznych występujących w systemie i zaprojektowanie odpowiednich mechanizmów zabezpieczających. Po zaprojektowaniu systemu przeprowadzana jest systemowa analiza ryzyka (*system hazard analysis*), której celem jest ocena bezpieczeństwa projektu systemu. Metoda drzew błędów stosowana jest też na poziomie analizy pojedynczych podsystemów (*subsystem hazard analysis*). Podczas eksploatacji systemu metoda FTA stosowana jest do oceny bezpieczeństwa, analizy wypadków oraz projektowania modyfikacji systemu.

Cele metody FTA obejmują [2]:

- identyfikację przyczyn (scenariuszy zdarzeń) powodujących wystąpienie stanów niebezpiecznych,
- określenie częstości ich występowania,
- określenie wymagań niezawodnościowych dla poszczególnych elementów systemu mających wpływ na występowanie stanów niebezpiecznych,
- określenie, czy niezawodność poszczególnych elementów systemu pozwala na osiągnięcie wyznaczonego poziomu

bezpieczeństwa w odniesieniu do danego stanu niebezpiecznego,
- wykazanie niezależności danego stanu niebezpiecznego od poszczególnych podsystemów,
- wykazanie wzajemnej niezależności awarii,
- identyfikację krytycznych elementów systemu,
- identyfikację błędów o wspólnych przyczynach mających wpływ na występowanie stanu niebezpiecznego.

Metoda drzew błędów składa się z kilku kroków:
- przygotowanie analizy:
  - definicja zakresu analizy,
  - zapoznanie się z definicją systemu, jego funkcjami, zasadami działania,
  - identyfikacja stanu niebezpiecznego;
- konstrukcja drzewa błędów;
- analiza drzewa i opracowanie wyników.

Przygotowanie analizy obejmuje zapoznanie się z konstrukcją systemu, definicję zakresu analizy oraz zdefiniowanie stanu niebezpiecznego, którego przyczyny będą analizowane. Definicja zakresu analizy obejmuje określenie poziomu szczegółowości, celu oraz założeń procesu analizy. Jawnie definiowany jest poziom szczegółowości dla poszczególnych podsystemów. Zakres analizy, uzależniony od stopnia zaawansowania projektu systemu, ma wpływ na czas i koszt jej wykonania. Założenia procesu analizy określają zakres wykorzystywanej wiedzy, przyjęte uogólnienia lub uproszczenia. W przypadku bardzo skomplikowanego systemu lub takiego, w którym nie są dokładnie znane zasady działania (np. przebieg reakcji w reaktorze), przyjmowany jest uproszczony model zachodzących procesów. Inną możliwością jest wykluczenie z procesu analizy pewnych procesów zachodzących w systemie. Elementy bierne (np. przewody elektryczne, rury) charakteryzują się niezawodnością o kilka rzędów wielkości większą niż elementy czynne (np. pompy i silniki). Wpływ elementów biernych na bezpieczeństwo systemu może być pomijalnie mały i pominięcie tego typu elementów pozwala na uproszczenie procesu analizy. Wszystkie tego rodzaju założenia powinny być udokumentowane i uzasadnione.

Budowa drzewa błędów wymaga znajomości dziedziny aplikacyjnej, zapoznania się z projektem systemu oraz technologią, w której jest on wykonany. Ostatnim krokiem przygotowania analizy jest identyfikacja stanu niebezpiecznego. Jest to jednocześnie pierwszy krok budowy drzewa błędów. Każdy znany stan niebezpieczny stanowi punkt wyjścia dla oddzielnego drzewa błędów. Jedno drzewo błędów opisuje przyczyny tylko jednego takiego stanu. Jest on nazywany zdarzeniem szczytowym drzewa błędów (*top-event*).

Podstawowe elementy drzewa błędów to zdarzenia (*events*) oraz bramy (*gates*). W metodzie FTA pojęcie zdarzenia jest bardzo szerokie. Zdarzeniem jest stan niebezpieczny, awaria elementu systemu, poprawne zadziałanie elementu czy wysłanie sygnału przez oprogramowanie. Bramy są wykorzystywane do opisu zależności przyczynowo-skutkowych pomiędzy zdarzeniami.

Drzewo błędów jest budowane poprzez cykliczne powtarzanie jednego kroku, jakim jest dodanie bezpośrednich przyczyn zdarzenia będącego liściem drzewa. Na początku analizy drzewo składa się z jednego tylko węzła, jakim jest zdarzenie szczytowe (*hazard*). Jest ono jednocześnie jedynym liściem w drzewie. W pierwszym kroku określane są więc bezpośrednie przyczyny powstania stanu niebezpiecznego. Zasada *bezpośredniego* związku przyczynowo-skutkowego jest bardzo istotna, gdyż zmniejsza możliwość pominięcia którejkolwiek z przyczyn i zapewnia lokalny charakter pojedynczego kroku konstrukcji drzewa.

W wyniku każdego kroku do drzewa zostają dodane nowe zdarzenia, będące przyczyną jednego z liści. Dodana też zostaje jedna brama łącząca to zdarzenie z jego przyczynami. Podstawowy krok konstrukcji drzewa błędów jest kontynuowany tak długo, jak długo możliwe jest określenie przyczyn kolejnych zdarzeń w ramach zdefiniowanego zakresu analizy.

Wyróżnia się kilka rodzajów bram definiujących logiczne zależności pomiędzy zdarzeniami. Podstawowe typy to AND (oznacza konieczność wystąpienia wszystkich przyczyn dla spowodowania zdarzenia wyjściowego) oraz OR (oznacza, że wystąpienie jednej z przyczyn wystarcza dla spowodowania zdarzenia wyjściowego). Przykład drzewa pokazano na rys. 2. Na szczycie drzewa znajduje się analizowane zdarzenie niebezpieczne – awaria systemu TRAIN A. Dla zdarzenia tego określono 4 przyczyny bezpośrednie. Każda z nich może spowodować awarię TRAIN A, więc połączone są symbolem bramy OR. Dla zdarzenia SIGNA zdefiniowano dwie przyczyny, które zawsze muszą wystąpić łącznie dla spowodowania niebezpiecznego zdarzenia. Zdarzenia te są więc połączone bramą typu AND. Pełen zestaw symboli stosowanych w drzewach błędów znajduje się w standardzie [2].

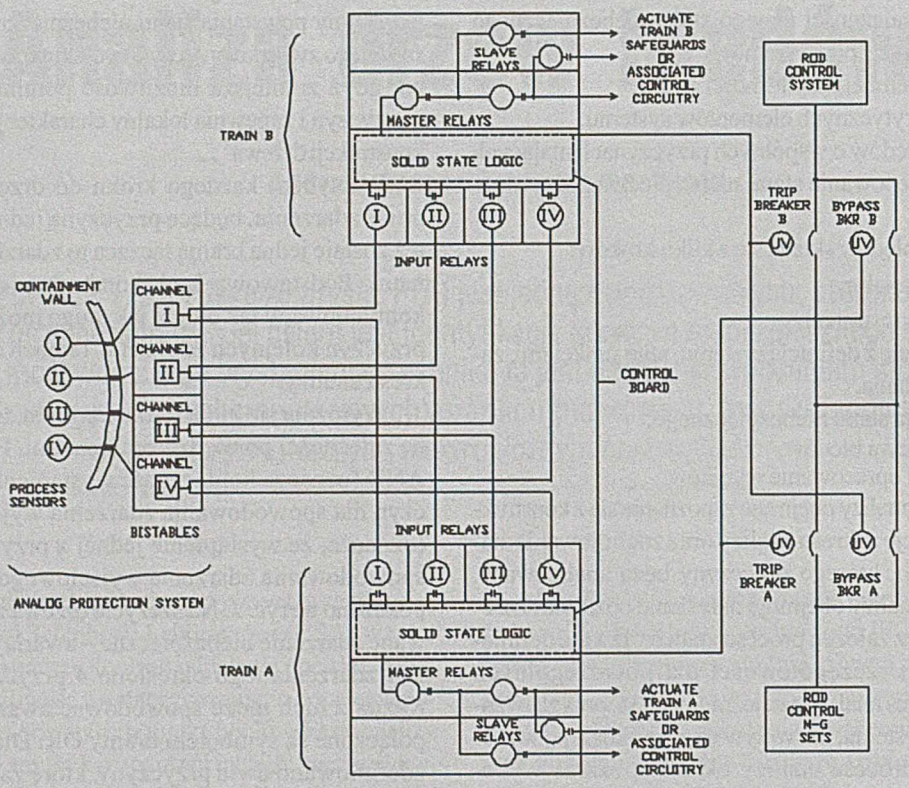
## Drzewo błędów systemu ESFAS

ESFAS (*Engineered Safety Features Actuation System*) jest systemem zabezpieczającym reaktor atomowy w elektrowni Krško (Słowenia) [3]. Jego zadaniem jest uruchomienie odpowiednich zabezpieczeń w chwili, gdy reaktor osiągnie stan niebezpieczny. Schemat fragmentu systemu pokazano na rys. 1.

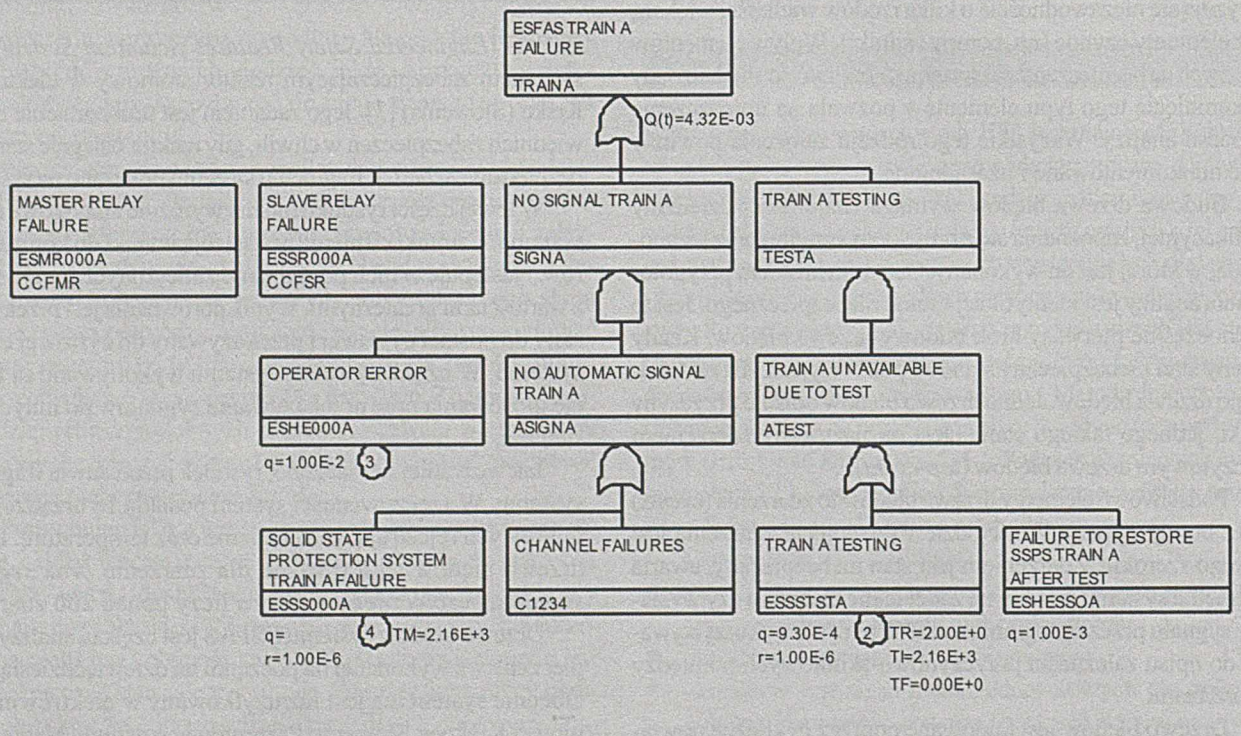
W lewej części rysunku można wyróżnić analogową część systemu. Czujniki ciśnienia przekazują wyniki do komparatorów. Następuje w nich porównanie zmierzonych parametrów z wartościami granicznymi. Wynik porównania jest przekształcany do postaci cyfrowej i przekazywany do cyfrowej części systemu. W części cyfrowej systemu wykonywane są funkcje głosowania oraz przekazywania sygnałów do innych systemów.

Jak wcześniej zaznaczono, rysunek przedstawia fragment systemu. W rzeczywistości system posiada 18 urządzeń pomiarowych rejestrujących ciśnienie oraz temperaturę. Pełne drzewo błędów tego systemu dla zdarzenia *brak sygnału stanu niebezpiecznego reaktora* liczy ponad 200 zdarzeń.

Drzewo błędów systemu ESFAS jest częścią analizy bezpieczeństwa wykonanej na początku lat dziewięćdziesiątych. Obecnie system ten jest już użytkowany w elektrowni atomowej Krško w Słowenii. Prezentowane drzewo błędów zostało wykonane przy pomocy narzędzia RISK SPEKTRUM. Przy jego użyciu wykonano też probabilistyczną analizę bezpieczeństwa.




Rys.1. Uproszczona struktura systemu ESFAS




Rys.2. Drzewo błędów systemu ESFAS

## Analiza minimalnych zbiorów przyczyn

Podstawową metodą analizy drzewa błędów jest określenie minimalnych zbiorów przyczyn (minimal cut sets). Celem jest określenie zdarzeń, które łącznie mogą spowodować wystąpienie stanu niebezpiecznego.

Minimalnym zbiorem przyczyn (oznaczanym dalej w skrócie m.z.p.) nazywany jest taki najmniejszy zbiór zdarzeń, dla którego wystąpienie wszystkich jego elementów powoduje wystąpienie stanu niebezpiecznego. Jednocześnie brak któregokolwiek z tych zdarzeń sprawia, że dany m.z.p. nie powoduje wystąpienia takiego stanu. Jeżeli drzewo błędów zostało poprawnie zbudowane, a więc zawiera wszystkie przyczyny pojawienia się stanu niebezpiecznego, to może on wystąpić tylko wtedy, gdy zajdą wszystkie zdarzenia z pewnego minimalnego zbioru przyczyn. Minimalne zbiory przyczyn są definiowane na podstawie prostych schematów wnioskowania opartych o logikę bram drzewa błędów. Dla drzewa błędów z rys. 2 można podać następujące minimalne zbiory przyczyn:

{ESMR00A}, {ESSR000A}, {ESHE00A, ESS000A}, {ESHE00A, C1234}, {ESSSTSTA}, {ESHESS0A}

Dla rzeczywistych drzew błędów, które zwykle liczą od 50 do 200 zdarzeń, minimalne zbiory przyczyn liczą zwykle więcej zdarzeń.

## Analiza probabilistyczna

Znajomość prawdopodobieństwa wystąpienia zdarzeń podstawowych drzewa błędów umożliwia określenie prawdopodobieństwa wystąpienia zdarzenia szczytowego drzewa. Analiza probabilistyczna obejmuje wyznaczenie:

- niedostępności systemu oznaczanej symbolem Qt (*system unavailability*), czyli prawdopodobieństwa zdarzenia szczytowego;
- udziału minimalnego zbioru przyczyn w niedostępności systemu (*fractional contribution*), wyznaczanego procentowo (patrz tabela 1);
- współczynnik zwiększenia ryzyka RIF (*risk increase factor*) dla zdarzenia i:
- współczynnik obniżenia ryzyka RDF (*risk decrease factor*) dla zdarzenia i:
- udziału w niedostępności systemu FC (*fractional contribution*) dla zdarzenia i:

RDF pozwala na zidentyfikowanie zdarzeń, dla których zmniejszenie częstości ich występowania spowoduje największy spadek niedostępności (czyli wzrost dostępności). Jeżeli zdarzenia podstawowe dotyczą awarii pojedynczych elementów systemu, pozwala to na identyfikację tych elementów, które powinny być w pierwszej kolejności zastąpione bardziej niezawodnymi odpowiednikami.

RIF pozwala na zidentyfikowanie zdarzeń, których częstsze występowanie powoduje wzrost niedostępności całego systemu. Elementy, których dotyczą zdarzenia o największym współczynniku, powinny być objęte najbardziej szczegółową obserwacją spośród wszystkich elementów systemu.

FC pozwala zidentyfikować elementy, których niedostępność w największym stopniu wpływa na niedostępność całego systemu.

## Analiza drzew błędów oprogramowania

Pod koniec lat 80-tych rozpoczęto prace badawcze nad stosowaniem metody drzew błędów do analizy oprogramowania. Oprogamowanie, w odróżnieniu od systemów mechanicznych, nigdy się nie psuje. Wszystkie awarie oprogramowania to błędy projektowe. Powoduje to, że analiza oprogramowania wymaga trochę innego podejścia. Powinno ono uwzględniać charakterystykę oprogramowania:

- obecność tylko błędów projektowych,
- duży stopień komplikacji,
- brak niezależności elementów,
- dowolność kształtowania struktury,
- łatwość wprowadzania zmian w kodzie programu,
- trudność testowania i weryfikacji.

W roku 1991 opublikowano opis metody SFTA (Software Fault Tree Analysis) [4]. Podczas konstrukcji drzewa określane są możliwe ścieżki dojścia do niebezpiecznego stanu oprogramowania. Dla każdej ścieżki prowadzącej do stanu niebezpiecznego ocenia się, czy jej wystąpienie jest możliwe. Jeżeli tak, wymagana jest modyfikacja programu. Analiza drzewa błędów daje wynik pozytywny, czyli stwierdza, że program nie powoduje stanu niebezpiecznego, jeżeli zostanie wykazane, że wszystkie ścieżki prowadzące do takiego stanu są niemożliwe.

Oprogramowanie składa się z dużej liczby prostych elementów będących instrukcjami języka programowania, takimi jak: przypisanie, wybór IF oraz CASE, pętle WHILE oraz FOR. Dla każdej konstrukcji opracowano szablon dla określenia błędu danej instrukcji. Szablon liczy jedną lub kilka bram. Takie szablony są tworzone dla wielu języków programowania, np. ADA [4] lub C/C++ [5]. Szablony obejmują wszystkie konstrukcje języka, w tym współbieżność w przypadku języka ADA.

Dla przykładu rozpatrzmy instrukcję przypisania. Błąd w takim przypadku może być spowodowany przez przypisanie wartości powodującej niebezpieczne zachowanie albo błąd wykonania samej operacji przypisania, co może obejmować błąd określenia wartości przypisywanej lub błąd obsługi wyjątku (exception). Szablon przedstawiono na rys. 3.

Szablony zawierają wszystkie możliwe przyczyny błędu danej konstrukcji języka programowania. Podczas analizy konkretnego programu należy stosować je w całości jako fragmenty drzewa błędów, a następnie sprawdzać, które zdarzenia są niemożliwe. Na przykład dla przypisania x := 5, gdzie x jest typu INTEGER, zdarzenie wyznaczenie wartości powoduje błąd znajdujące się w szablonie jest niemożliwe. Dalszej analizie podlegają wtedy dwa pozostałe zdarzenia, to znaczy zmiana wartości powoduje błąd oraz przypisanie do złej lokacji pamięci.

## Integracja analizy oprogramowania oraz systemu

Zakres zastosowania metody SFTA obejmuje oprogramowanie, nie stosuje się jej natomiast do analizy całego systemu

**Rys.3. Szablon dla instrukcji przypisania**

(jego części mechanicznej wraz z oprogramowaniem). Standardowa procedura postępowania jest następująca: tworzone jest systemowe drzewo błędów w tradycyjny sposób. Niektóre zdarzenia podstawowe tego drzewa dotyczą błędów oprogramowania. Dla tych zdarzeń tworzone są osobne poddrzewa metodą SFTA.

Podobne podejście zastosowano w przypadku systemu Robertino [6, 7]. Jest to tworzony obecnie sterowany komputerowo robot o 4 stopniach swobody, stosowany do prac konserwacyjnych w reaktorze fuzyjnym (*thermo-nuclear fusion reactor*). Do chwili obecnej wykonano model w skali 1/3. Model ten ma wymiary 2,2m x 3m x 6,5m i jest w stanie przenosić ciężary o wadze do 6,5 tony z dokładnością do 0,1 mm. Model ten służy do testów przed zastosowaniem systemu w prawdziwym reaktorze. Rzeczywisty robot ma ponad 15 metrów wysokości i udźwig do 135 ton [8]. Brak precyzji ruchu tak dużych ciężarów może spowodować uszkodzenie reaktora. Robot obecnie stosowany w reaktorze jest sterowany ręcznie.

Do analizy bezpieczeństwa systemu Robertino zastosowano metody FTA i FMEA [8]. Analiza bezpieczeństwa była podzielona na 3 etapy:

1. identyfikacja stanów niebezpiecznych (w tym analiza wypadków) oraz określenie zakresu analizy,
2. wstępna analiza stanów niebezpiecznych oraz konstrukcja systemowych drzew błędów wysokiego poziomu,
3. analiza FMEA oraz konstrukcja drzew błędów dla zdarzeń podstawowych drzew błędów wysokiego poziomu.

Oprogramowanie podlega analizie w etapie trzecim.

Jak wskazano podczas omawiania tradycyjnej metody drzew błędów, głównym rodzajem analizy jest analiza probablistyczna. Gdy taka analiza połączona jest z analizą oprogramowania, pojawia się następujący problem: jakie jest prawdopodobieństwo wystąpienia tych zdarzeń drzewa błędów, które są awariami oprogramowania? W wielu analizach bezpieczeństwa, prowadzonych głównie w latach osiemdziesiątych, zakładano, że ponieważ oprogramowanie się nie psuje, to prawdopodobieństwo jego awarii jest równe 0. W przypadku systemu medycznego Therac-25 przyjęto, że prawdopodobieństwo awarii oprogramowania w czasie jednej godziny pracy wynosi $10^{-4}$. Taką właśnie wartość wpisano do drzew błędów dla wszystkich zdarzeń związnych z oprogramowaniem. Wartość ta nie była jednak zgodna z rzeczywistością. Błędy oprogramowania tego systemu spowodowały śmierć 6 osób w latach 1985-1987 [9].

W niedługim czasie w życie ma wejść standard IEC 1508, który będzie w znacznym stopniu regulować zagadnienia bezpieczeństwa oprogramowania. Definiuje on poziomy bezpieczeństwa wraz z prawdopodobieństwem awarii dla każdego z nich [1]. Standard zapewnia, że na najwyższym poziomie bezpieczeństwa prawdopodobieństwo awarii oprogramowania nie będzie większe niż $10^{-4}$/rok, co oznacza około $10^{-8}$/godz. Ocenia się, że jest to granica możliwości obecnej technologii wytwarzania oprogramowania [10].

❖ ❖ ❖

Analiza drzew błędów jest powszechnie stosowaną metodą analizy bezpieczeństwa z powodu wysokiej efektywności, prostoty oraz łatwości stosowania. Drzewa błędów, ze swej natury, pozwalają na koncentrację uwagi na awariach systemu zagrażających bezpieczeństwu. Graficzna reprezentacja drzewa ułatwia zarządzanie analizą oraz niezależną ocenę. Drzewa mogą być łatwo łączone lub dzielone, ich modyfikacje pozwalają na analizę wariantów konstrucji systemu. Obecnie na rynku dostępnych jest szereg narzędzi programowych wspierających analizę drzew błędów. Pozwalają one na efektywną analizę probabilistyczną złożonych systemów.

Obecnie metoda FTA nie pozwala na zintegrowaną analizę bezpieczeństwa oprogramowania wraz z systemami wykonanymi w innych technologiach. Tradycyjne drzewa błędów są stosowane do analizy systemowej, a SFTA do analizy samego oprogramowania. Zintegrowana analiza jest obecnie przedmiotem badań naukowych [11, 12 TN96, W96].

**Literatura**

[1] ***Functional safety: safety-related systems***, International Electrotechnical Comission standard draft, IEC 1508, 1994

[2] ***Fault Tree Analysis (FTA)***, International Electrotechnical Commission, International Standard 1025, 1990

[3] Čepin M., Kožuh M., Mavko B.: ***Safeguards Actuation Fault Tree Analysis***, International Conference on Probabilistic Safety Assessment and Management PASM-II, San Diego, March 20-24, 1994

[4] Leveson Nancy G., Cha S. S., Shimeall T. J.: ***Safety Verification of ADA programs using Software Fault Trees***, IEEE Software, lipiec 1991

[5] Voas Jeffrey M., Miller Keith W.: ***A Statistical and Automated Code-Based Fault-Tree Mitigation Framework for C++***, Proc. 1st Annual ENCRESS Conference Safety and Reliability of Software Based Systems, Bruges (Belgium), September 12–15, 1995

[6] Maier Thomas, Wilikens Marc: ***Reliability and Safety of Robotics: Aspects Related to Remote Handling, Part 1: System Safety Analysis and Software Safety Requirements for a Remote Handling Device***, Joint Research Centre of the EC, Institute for System Engineering and Informatics, Ispra (Włochy), Technical Note I.94.130 ISEI/IE 2784/94, sierpień 1994

[7] Morales Emilio Ruiz: ***A Software Development Approach for Robotics Control Systems***, SAFECOMP'95, strony 317-330

[8] Maier Thomas: ***FMEA and FTA to Support Safe Design of Embeded Software in Safety-Critical Systems***, Proc. 1st Annual ENCRESS Conference Safety and Reliability of Software Based Systems, Bruges (Belgium), September 12–15, 1995

[9] Leveson Nancy G., Turner Clark S.: ***An Investigation of the Therac-25 Accidents***, IEEE Computer, July 1993, strony 19-41

[10] Littlewood Bev, Wright David: ***A Bayesian Model that Combines Disparate Evidence for the Quantitative Assessment of System Dependability***, 14th International Conference on Computer System Safety, Reliability and Security SAFECOMP'95, Belgirate (Włochy), 11-13 października 1995, strony 173-188

[11] Trafford Paul, Norrie Kate: ***The Provision of Safety Requirements from Fault Trees and their Validation in Formal Models***, Research Report KUCSES96-05, Kingston University (UK), grudzień 1996

[12] Wardziński Andrzej: ***Analiza drzew błędów systemów komputerowych związanych z bezpieczeństwem***, Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki, rozprawa doktorska ETI 6/96, Gdańsk, 1996.

**Andrzej Wardziński** jest pracownikiem Centrum Inżynierii Oprogramowania.
**Marko Čepin** jest pracownikiem Instytutu Josefa Stefana z Lubljany.

Dział Systemy Czasu Rzeczywistego jest dofinansowany przez Komitet Badań Naukowych.

# Fatamorgana, mechanizmy napędowe i korzenie informatyki

## (ujęcie historyczne od samego początku)[1]

Zygmunt Ryznar

## Fatamorgana

Informatyka zastosowań wydaje się być dziedziną bardzo konkretną, gdyż wszystko tutaj w ostatecznym przypadku sprowadza się do programów i danych. Niemniej jednak, poza teorią i praktyką projektowania i programowania systemów, istnieje w informatyce znaczny obszar dedykowany zagadnieniom ogólnym, dookoła których toczy się tzw. „wieczna dyskusja". Niektóre tematy dyskusji w sposób naturalny zanikają – jak fatamorgana – gdy po pewnym czasie okazuje się, iż nie można wypełnić je konkretną treścią, niektóre natomiast wywoływane są przedwcześnie i muszą poczekać aż nadejdzie odpowiednie wsparcie techniczne i technologiczne, a jeszcze inne typu „wyższości świąt Bożego Narodzenia nad Wielkanocą" (np. języka PL/1 nad COBOLem) nigdy nie doczekają się rozstrzygnięcia (chyba, że za takie uznamy – w przypadku języków programowania – liczbę kompilatorów i programistów z nich korzystających).

Do tematów przedwczesnych zaliczyć można sprawę systemów ZSIK (Zintegrowanych Systemów Informowania Kierownictwa, czyli **IMIS** – Integrated Management Information Systems), który to temat został na przełomie lat 60-tych i 70-tych przedwcześnie „rozdmuchany" na dużą skalę międzynarodową przez organizację Diebold. Odbyło się wiele światowych konferencji, na których interes finansowy robili organizatorzy, natomiast uczestnicy podlegali „ideologicznemu praniu mózgów" jak „integrować wszystko ze wszystkim" aby system zasługiwał na miano nowoczesnego. Odbywało się to w sytuacji, kiedy nie było odpowiednich systemów zarządzania wielkimi bazami danych, istniały tylko komputery mainframe (kosztujące fortunę), brakowało sprawnych systemów sieciowych i architektury klient-serwer, interfejsu typu GUI, technologii hurtowni danych, nikomu nie śniło się o systemach ekspertowych opartych na bazach wiedzy itp. Nic dziwnego, że projekty takich systemów – mimo iż prowadzone przez wielkie firmy i przy dużych nakładach – skazane były z góry na porażkę.

Ale nawet teraz, często nie udaje się uniknąć **syndromu wielkich systemów informatycznych,** polegającego na tym, że łatwo jest zacząć projektowanie systemu kompleksowego, lecz nie sposób skończyć. Użyteczne może być stosowanie zasady **TBSS** (*„Think Big Start Small"*), gdyż łączy ona w sobie podejście kompleksowe w skali instytucji (enterprise view) z sukcesywną (zwaną w literaturze angielskiej *„discrete implementation"* lub *„rapid appplication"*) – budowy systemu poprzez poszczególne moduły o dobrze zdefiniowanych granicach. Praktyczną realizacją *„think big"* jest sporządzenie globalnego modelu systemu (odwzorowującego istotę i specyfikę biznesu danego przedsiębiorstwa poprzez zdefiniowanie klas obiektów i procesów) oraz sukcesywne zakładanie repozytorium obiektowo zbudowanych procedur i funkcji.

Bardzo ciekawym *epizodem historycznym* informatyki światowej były kilkuletnie, prowadzone na początku lat 80-tych, prace badawcze mające na celu sformułowanie języka specyfikacyjnego do odwzorowywania świata biznesu prowadzone pod egidą komitetu X3/SPARC, mające być podstawą tzw. schematu konceptualnego. Byłby to na pewno dobry przyczynek do budowy systemów zorientowanych na kierownictwo, gdyby doprowadził do konkretnych wyników (np. międzynarodowych standardów) i odpowiedni pomost do prac nad budową „automatycznych fabryk oprogramowania" (Automatic Software Factory), bazujących na specyfikacjach biznesowych, następnie odzwzorowujących

[1] Należę do tzw. seniorów informatyki. Pierwszy program w języku wewnętrznym napisałem w 1957 roku. Potem „marnowałem" energię na oprogramowywanie kolejnych wynalazków typu ODRA 1003, ZAM41, Riad 32, komputerów Honeywella, minikomputera Seecheck, czy wreszcie kochanego przez wszystkich Peceta. Ostatnio dotykałem minikomputerów ALPHA. Ale przede wszystkim zajmowałem się problematyką baz danych, tworzyłem aplikacje w różnych dziedzinach. Ostatnie parę lat poświęciłem informatyce bankowej, co pozwoliło mi wreszcie zrozumieć zawiłości złożonych systemów działających w czasie rzeczywistym. W sumie działam więc w informatyce 40 lat, co w zakresie podsumowań daje pewną przewagę w stosunku do najmłodszego pokolenia informatyków oraz stwarza nadzieję, iż moje „historyczne" rozważania dostarczą troszkę interesującej wiedzy Czytelnikom czasopisma Informatyka, w którym dawno dawno temu debiutowałem (w latach 60-tych, nazywało się wtedy „Maszyny Matematyczne").

je na poziom logiczny (niezależny od platformy komputerowej) i wreszcie na poziom fizyczny (implementacyjny).

## Mechanizmy napędowe informatyki

Wiadomo, że informatyka stanowi siłę napędową wielu dziedzin gospodarczych, zarówno cywilnych jak i wojskowych. Komputery i mikroprocesory występują dosłownie wszędzie. Nie da się bez nich ani prowadzić sprawnie biznesu (np. bankowego), ani kierować samolotem lub nowoczesnym samochodem, sterować siecią telekomunikacyjną itp.

W gospodarce rynkowej komputery występują jako czynnik wzmacniający konkurencyjność poprzez szybkość obsługi, jakość produktów, zapewnienie klientowi komfortu obsługi (np. poprzez internetowy home/office banking), a wreszcie zmniejszenie pracochłonności produkcji czy usług.

*Warto jednak zastanowić się co stanowiło (i stanowi) siłę napędową samej informatyki,* co spowodowało, iż dziedzina ta rozwijała się tak dynamicznie jak żadna inna. Świadczą o tym m.in. następujące dane statystyczne: w 1951 roku było na świecie 100 komputerów, zaś 20 lat później już 1000 krotnie więcej (czyli 100 tysięcy). Teraz ilość sprzętu komputerowego (szczególnie komputerów osobistych) jest praktycznie niemierzalna oraz trudno znaleźć dziedziny, w których komputery nie są stosowane.

*Ale nawet teraz, często nie udaje się uniknąć syndromu wielkich systemów informatycznych, polegającego na tym, że łatwo jest zacząć projektowanie systemu kompleksowego, lecz nie sposób skończyć.*

Wśród mechanizmów napędowych wymienić można potrzeby, działania rządowe, czasem wydarzenia dziejowe, walkę konkurencyjną, modę na komputery osobiste. Niepoślednią rolę w rozwoju informatyki odegrały indywidualności ze środowisk nauki i biznesu.

Napewno główną siłą napędową sa potrzeby, które początkowo polegały przede wszystkim na wspomaganiu obliczeń naukowych, statystycznych i technicznych na uczelniach i w biurach projektowych, oraz sterowaniu procesami technologicznymi. Później rozpoczęła się era zastosowań komputerów w obsłudze biznesu, a więc tzw. przetwarzanie danych, w którym niewydolne okazały się maszyny licząco-analityczne oparte na technologii kart dziurkowanych. Gwałtowny wzrost zastosowań komputerow nastapił w drugiej połowie lat 60-tych, kiedy istniało już ponad 1000 specjalizowanych języków programowania, powszechnie stosowano pamięci dyskowe, itp.

Ciągły wzrost wolumenu danych ewidencyjno-finansowych prowadził do rozbudowy pamięci komputerów i urządzeń peryferyjnych (wejścia-wyjścia), a w ostatnich latach doprowadził do znacznych zmian technologii przetwarzania danych (m.in. poprzez wprowadzenie wielowymiarowych hurtowni danych).

Duże zasługi dla rozwoju informatyki wniósł rząd USA. To właśnie dzięki jego zamówieniom rozwinął się amerykański przemysł komputerowy i rozpoczęto prace standaryzacyjne, porządkujące walkę konkurencyjną.

Wśród wydarzeń dziejowych wymienić należy fakt, iż pod wpływem potrzeb II wojny światowej rozpoczęto prace nad pierwszym komputerem elektronicznym (lampowym) ENIAC, który miał służyć (nie zdążył) do obliczeń torów balistycznych pocisków. Zapotrzebowanie na superszybkie komputery istniało dalej w związku z pracami nad bombą atomową, a potem lotami kosmicznymi.

Interesującym tematem badawczym, np. dla socjologów, może być analiza wpływu kultury (technicznej, ogólnej) społeczeństwa oraz systemu społeczno-ustrojowego na rozwój informatyki, w tym stworzenie sprzyjających warunków dla działalności jednostek wybitnych .

Pojawiają się tu interesujące pytania, np.: w jakim stopniu kultura ogólna społeczeństwa sprzyja innowacyjności oraz czy taką rolę spełnia dobrobyt, gdyż wiadomo, iż tzw. pasywny dobrobyt prowadzi w zasadzie tylko do konsumpcji.

Odrębnego potraktowania wymaga amerykański fenomen self-made-mana, a więc przysłowiowych karier „od czyścibuta do milionera", zdarzających się w każdej dziedzinie, a więc i w informatyce, a możliwych w systemie, który ceni jednostki.

System PRL'owski i społeczeństwo polskie „nie lubiło" (i zapewne nie do końca zostało to przełamane) ludzi wychylających się, posiadających własne zdanie i własne inicjatywy ....(lepsi byli Ci „bierni ale wierni"). Casus Jacka Karpińskiego na początku lat 80-tych – zdolnego (a może nawet wybitnego) konstruktora mikrokomputerow K-202, któremu tak obcinano skrzydła aż zajął się „hodowlą trzody chlewnej". W systemie tym również wysiłki typowo instytucjonalne (np. krakowskiej fabryki KFAP w zakresie podjęcia przemysłowej produkcji mikrokomputerów KRAK) „paliły na panewce".

W odróżnieniu od tych przykładów, powstanie większości największych firm komputerowych w USA było i jest wynikiem działania wybitnych lub kreatywnych jednostek. Firma HP została założona w 1938 roku w garażu z kapitałem 538 USD przez Davida Packarda i Williama Hewletta, a teraz obraca miliardami dolarów i zatrudnia ponad 100 tysięcy osób.

W 1977 roku Larry Elison wraz z Robertem Minerem założyl firmę Oracle z kapitałem założycielskim wynoszącym zaledwie 1200 USD.Twórcy Microsoftu B.Gates i Paul Allen siedzieli od początku blisko mikroprocesorów i wiedzieli „kiedy, gdzie i co" może być potrzebne w związku z ich stosowaniem. Już w 1974 roku kupili jeden z pierwszych mikroprocesorów 8008, a następnie opracowali język BASIC dla komputera z procesorem 8080. Potem zaś prowadzili prace nad systemem operacyjnym dla IBM'owskiego prapeceta z procesorem 8088.

Firma Intel została założona w 1968 r. – przy bardzo ni-

skim kapitale własnym – przez grupę inżynierów (w tym ojca pierwszego układu scalonego Boba Noyce'a), którzy porzucili firmę Fairchild Semiconductor.

To wiemy o tych, którzy tworzyli największy biznes komputerowy stosunkowo niedawno. Myślę, że równie ciekawe są informacje o jednostkach, które kiedyś stworzyły teoretyczne podwaliny pod powstanie samej informatyki[2].

## Korzenie informatyki

### *Zaczęło się w Anglii*

***- Charles Babbage (1792-1871)***

Kierował w latach 1828-1839 katedrą matematyki w Cambridge. Sześćdziesiąt lat swojego życia poświęcił koncepcji i budowie fenomenalnych jak na XIX wiek mechanicznych, automatycznych maszyn liczących (pamięć czasowa i długotrwała, urządzenie arytmetyczne, wejście z kart dziurkowanych – znanych wówczas z maszyn tkackich, instrukcje działań warunkowych itp.

***- George Boole (1815-1864)***

Stworzył algebrę logiki, która znalazła potem zastosowanie w teorii komputerów.

***- Allan M.Turing***

W 1936 roku w odczycie wygłoszonym w Londyńskim Towarzystwie Matematycznym przedstawił teoretyczny model abstrakcyjnego urządzenia do przekształcania symboli, nazwanego potem maszyną Turinga .

### *USA ojczyzną komputerów*

***- John von Neumann (1903-1957)***

Amerykański matematyk węgierskiego pochodzenia. Był wszechstronnym uczonym na miarę epoki renesansu i wniósł duży wkład w rozwój takich dziedzin jak mechanika kwantowa, logika matematyczna, topologia i teoria gier. Był profesorem na Uniwersytecie w Princeton i Institute of Advanced Studies, członkiem komisji d/s energii atomowej.

Już w 1936 roku, niezależnie od Francuza Couffignala, zaproponował zastosowanie dwójkowego systemu liczenia w maszynach liczących. Pracował nad ulepszeniem pierwszego komputera elektronicznego ENIAC. W serii raportów publikowanych w latach 1945-47 podał fundamentalne zasady budowy maszyn matematycznych (w systemie binarnym można przedstawiać zarówno dane jak i program, ponieważ rozkazy są kodowane i przechowywane tak samo jak liczby, można je przetwarzać arytmetycznie w celu otrzymania nowych rozkazów, program powinien być przechowywany w pamięci itp.).

[2] Przypomnienie w tym artykule zasług i faktów historycznych, związanych z nazwiskami, niech będzie wyrazem pewnego hołdu złożonego pionierom, o których nie zawsze pamiętają twórcy encyklopedii.

Ze strony polskiej wymienić trzeba koniecznie logika Jana Łukasiewicza (1878-1956), twórcę koncepcji beznawiasowego zapisu wzorów algebraicznych, zwanej zapisem polskim (polish notation, reverse polish notation), wykorzystywanego szeroko w technice kompilacji.

Jak widzimy, oczywiste dzisiaj zasady działania komputerów odkrywane były przez lata, a nawet wieki.

### *Konkurencja pierwszych języków programowania*

Jako trzy pierwsze języki programowania znane są:

**FORTRAN–FORmula TRANslator (1954-5),**
**ALGOL–ALGOrithmic Language (1958-1960),**
**COBOL–Common Business Oriented Language (1960).**

FORTRAN, który właściwie był językiem firmowym IBM, do 1961 r. był ignorowany przez innych producentów. Z kolei IBM próbował bojkotować język COBOL, przeciwstawiając mu swój Autocoder i Commercial Translator. Podobną postawę w stosunku do COBOL'u zajął początkowo Honeywell, prowadząc prace nad własnym językiem FACT. Dopiero ogłoszona w 1964 roku przez rząd USA blokada zakupu komputerów nie wyposażonych w translator języka COBOL oraz konkursowe zakupy komputerów wyposażonych w najlepsze kompilatory języka COBOL, spowodowały utrwalenie pozycji tego języka. Rząd amerykański miał w ręku istotne argumenty, dokonując bardzo znacznych zakupów (np. w 1963 roku tylko lotnictwo wojskowe Air Force zakupiło 500 komputerów).

Język ALGOL nie miał szczęścia do sponsorów rządowych i firm komputerowych, w związku z czym – mimo dużej popularności w środowisku naukowym – powoli zanikał. Jest przykładem produktu organizacji naukowych takich jak GAMM (Geselschaft fur Angewandte Mathematik und Mechanik), ACM (Association for Computing Machinery) oraz organizacji użytkowników takich jak British Computer Society, USE, SHARE itd. Ostatnim znaczącym wkładem w rozwój tego języka jest chyba ALGOL68 zdefiniowany przez grupę roboczą 2.1. organizacji IFIP (International Federation for Information Processing).

Przykładem dynamicznej ekspansji jest język C, który powstał w Bell Labs w grupie (uczestniczył w niej D.Ritchie) pracującej nad UNIXem . Poprzedzały go języki B, NB na PDP-11. Od 1973 roku pisany jest w nim system operacyjny UNIX. Ostatnio obserwuje się, np. w USA, zjawisko opracowywania w tym języku (C, C++, Visual C++) wielu pakietów aplikacyjnych (np. bankowych). Decyduje widać potrzeba tworzenia systemów z bardziej efektywnym przetwarzaniem, zaś stało się to możliwe dzięki wykształceniu dostatecznego potencjału programistów w języku C.

Koncepcja uniwersalnego języka pośredniego UNCOL (Universal Computer Oriented Language) powstała w latach 1958-1961 w artykułach M.E. Convey'a., T.B. Steel'a. i J. Stronga, lecz niestety nie doszło do jego opracowania. Miał to być język pośredni dla wszystkich platform sprzętowych, na który miały być tłumaczone inne języki programowania. Miało to zmniejszyć liczbę kompilatorów, ale

w istocie swojej był to zalążek pełnej otwartości (jakby uniwersalne API), być może coś takiego do czego dążą zwolennicy języka JAVA.

### „Ślepe uliczki" informatyki

Informatyka rozwija się bardzo szybko, nie zawsze jednak tak jak przewidywano. Oto przykłady rozbieżności:

- w 1958 r. zbudowano pierwszy perceptron czyli urządzenie do rozpoznawania pisma oraz dźwięków, tymczasem dopiero po 35 latach od tego wydarzenia pojawiły się komputery multimedialne,
- w 1959 roku zastosowano kriotriony w charakterze pamięci operacyjnej i przewidywano, że zastąpią one pamięć ferrytową,
- na początku lat 70-tych eksperci firmy Rand Corp. twierdzili, że niedługo po roku 1990 nastapi bezpośrednie podłączenie komputera do mózgu człowieka, zaś jeszcze do 2000 roku powstanie możliwość uczenia się przez bezpośrednie – za pomocą komputera – utrwalanie informacji w korze mózgowej !

*Kraje takie jak Polska stały się rynkiem zbytu dla potentatów zachodnich i nie wykorzystują własnego potencjału twórczego, chociaż bywają chlubne wyjątki.*

- w pierwszych latach 60-tych pojawiły się koncepcje maszyn bezadresowych. Po pierwszych eksperymentach w Polsce i Australii nie doszło do przemysłowej produkcji tych maszyn.

Zwolennicy języka PL/1 (stworzonego w połowie lat 60-tych dla serii IBM/360) przewidywali, że stanie się on językiem uniwersalnym, zaś zwolennicy języka RPG (Report Program Generator) w 1967 roku twierdzili, że aż 75% wszystkich programów przetwarzania danych zostanie napisanych w tym języku, podczas gdy tylko 15% przypadnie na COBOL. Tymczasem język COBOL utrzymał swoją pozycję – szczególnie na komputerach typu mainframe – jako język tzw. przetwarzania danych, aczkolwiek ostatnio traci bardzo na rzecz języka C.

### Stany Zjednoczone wiodą prym

Powyższe stwierdzenie jest prawie takim samym truizmem jak amerykański dyktat mody filmowej. Nie wszyscy jednak wiedzą, że ta hegemonia komputerowa dotyczy nie tylko sprzętu (typu mainframe i minikomputery), oprogramowania systemowo-narzędziowego (systemy operacyjne, systemy zarządzania bazami danych, wielowymiarowe hurtownie danych, narzędzia typu CASE, przeglądarki internetowe), lecz również oprogramowania aplikacyjnego.

Można to udowodnić na przykładzie oprogramowania dla bankowości, zaliczanego do najtrudniejszych ze względu na złożoność instrumentów finansowych stosowanych na rynkach kapitałowych (nie mam tutaj na myśli stosunkowo prostych lokat i kredytów), znaczne obciążenie zarówno trybem czasu rzeczywistego jak i przetwarzania w partiach, oraz rygorystyczne wymogi bezpieczeństwa. Na 250 systemów i pakietów bankowych istniejących na międzynarodowym rynku ponad 50% (130) pochodzi z USA, 20 z W. Brytanii i 12 z Niemiec. Miastami, w których tworzy się najwięcej tego typu oprogramowania, są Nowy Jork i Londyn.

W zastosowaniach przemysłowych Europa trzyma się dzięki parze niemieckich domów software'owych SAP i Software AG, przy czym należy tutaj podkreślić ekspansję firmy SAP na rynku amerykańskim, gdzie osiąga obroty takie jak w Niemczech!

Twórcy oprogramowania aplikacyjnego coraz mniej związują się z pojedynczymi producentami komputerów (kiedyś większość z nich tworzyła systemy ukierunkowane tylko na mainframe'y IBM, AS/400 IBM, albo Vaxy Digitala) . W latach 90-tych powstało kilkadziesiąt systemów bankowych, które mogą być eksploatowane zarówno na HP9000 jak i na ALPHAch Digital'a, a czasem również na RS6000 IBM'a lub Sun- SPARC Sun'a.

### Najważniejsze daty w zakresie postępu technicznego informatyki

| | |
|---|---|
| 1947 | zastosowanie bębna magnetycznego w charakterze pamięci |
| 1954 | pierwsze zastosowanie tranzystorów w komputerze (TRADIC) |
| 1954 | pierwsze użycie taśmy magnetycznej (UNIVAC I) |
| 1955-56 | pierwsze zastosowania pamięci ferrytowych (IBM 704,705, Remington Rand w UNIVAC 1103A) |
| 1961 | IBM wyprodukował pierwszy dysk magnetyczny |
| 1971 | pierwszy mikroprocesor – w postaci układu scalonego (firma intel 4bitowy 4004) |
| 1974 | pierwszy mikroprocesor 8-bitowy ( 8008) |
| 1981 | premiera IBM'owskiego PC z procesorem 8088 uważana za narodziny komputera osobistego. |

### *Cień monopolu*

Prawie od początku przemysł komputerowy był zdominowany przez firmę IBM, zarówno pod względem obrotów jak i zatrudnienia. Pozycja finansowa IBM jest nadal niepodważalna (1996 – obroty 76 mld dolarów, HP – 31, Digital – 15, Microsoft – 8,7). Przemysł komputerowy to bardzo ryzykowny interes, pozostawiający na „placu boju" tylko „najsilniejsze armie". Kto teraz pamięta o takich zasłużonych firmach komputerowych jak Sperry/Remington Rand, GE (General Electric), Honeywell, Ferranti ... a na krajowym podwórku ELWRO.

Na szczęście, praktykom monopolistcznym w USA przeciwdziała Ministerstwo Sprawiedliwości (Department of Justice). Można tutaj wymienić sprawę zablokowania kontraktu zakupu firmy Intuit (twórcy Quicken) przez Microsoft za 2,1 mld dolarów jako próby zmonopolizowania rynku „personal finance software".

Pojawienie się na rynku wydajnych i o wiele tańszych od mainframe'ów minikomputerów oraz architektury klient-serwer (preferującej tzw. „down-sizing") doprowadziło do zmiany strategii technologicznej firmy IBM . Po 1993 roku firma ta zmieniła technologię mainframe'ów produkowanych dotychczas w technologii ECL (emitter-coupled logic) na ulepszoną mikroprocesorową technologię CMOS (Complementary metal oxide semiconductor). Komputery stały się wolniejsze (układy ECL były co najmniej 3 krotnie szybsze od zwykłych CMOSowych), ale za to tańsze. Wkład IBM w ulepszenie technologii CMOS był znaczny (nastąpił wzrost ich szybkości od 1,3 MIPS do 40 MIPS). Przewiduje się, że do roku 1998 ulepszone procesory CMOS osiągną szybkość porównywalną z klasycznymi procesorami ECL (poza konkurencją pozostaną prawdopodobnie hybrydowe procesory produkowane przez Hitachi).

## Przełomowe wydarzenia w historii oprogramowania

### *Rok 1949*

– Amerykanin *Wilkes* niezależnie od Neumanna wykazał, że należy dokonywać operacji arytmetycznych na adresach, co pozwala na tworzenie pętli w programie (zmniejszając jego wielkość).

### *Rok 1952*

– Heinz Ruthishauser (z Federalnego Instytutu Technologicznego w Zurichu) przedstawił metodę wprowadzania do komputera wyrażeń algebraicznych w zwyczajnej formie i zaproponował zwięzłą instrukcję do organizacji pętli „for k = i .."

– zastosowanie matematycznej notacji SHORT CODE zaproponowanej przez Mauchley'a (współtwórcę ENIACa) w maszynach UNIVAC i BINAC

– Grace Hopper zatrudniona w Remington Rand jako szef działu programowania wprowadziła termin „kompilator" i zaproponowała użycie języka angielskiego do pisania programów, a następnie opracowała prawdopodobnie pierwsze w świecie kompilatory A0, A1.

### *Rok 1953*

– dla maszyny IBM 701 opracowano Speed Coding System w opracowaniu którego uczestniczył J.Backus (późniejszy – 1959 – autor notacji BNF stosowanej w opisie ALGOLu).

– na jednym z pierwszych posiedzeń ACM (Association for Computing Machinery) omawiano sprawę tzw. prymitywistów (zwolenników języka wewnętrznego) oraz „żołnierzy postępu" (space cadets). Ci ostatni posądzani byli o propagowanie luksusu oprogramowania wg „śmiesznych" schematów ograniczających myślenie.

Przy okazji tych zmian warto zwrócić uwagę na to, że równolegle do „down-sizing" przebiega proces „up-sizing", polegający na rozbudowie minikomputerów (typu HP9000, Digital ALPHA, Sun Ultra Enterprise) do granic wydajności mainframe'ów. Można to było osiągnąć poprzez architektury, wieloprocesorowość, przetwarzanie wielowątkowe, adresowanie 64-bitowe i coraz szybsze procesory. Wygląda więc na to, że historia się powtarza i powrócimy do stanu wyjściowego. Kolejnym tematem po architekturze klient-serwer, może być bitwa o opanowanie internetowego biznesu, szczególnie w zakresie płatności finansowych. W tej konkurencji wystartował już Microsoft ze swoim Microsoft Merchant System.

O pozycji na rynku i tzw. „słuszności" rozwiązania decyduje przede wszystkim siła finansowa firmy oraz skuteczna reklama (dotarcie do wszystkich potencjalnych klientów). Początkowy burzliwy rozwój twórczy (wielu producentów w wielu krajach) został zastąpiony przez powielanie tego samego – niekoniecznie najlepszego – rozwiązania w ogromnej skali (np. klasycznego PeCeta opartego na 640 KB bazowej pamięci operacyjnej). Producenci oprogramowania systemowego i narzędziowego napędzają klientów producentom sprzętowym poprzez coraz bardziej pamięciożerny i spowalniający pracę software. Co prawda i na szczęście, nie towarzyszy temu proporcjonalny wzrost cen, ale tempo starzenia moralnego sprzętu i oprogramowania jest tak duże, że klienci wydają na komputeryzację coraz więcej pieniędzy.

Niezależnie od wydatków finansowych, popadają oni w coraz większą zależność od monopolistów (przeważnie z dalekiego kraju) i abstrakcyjną (w sytuacji gdy nie ma wyboru) staje się kwestia reklamowania jakości oprogramowania, np. gdy przy znacznym obłożeniu system „pada" kilka razy dziennie . Kraje takie jak

Polska stały się rynkiem zbytu dla potentatów zachodnich i nie wykorzystują własnego potencjału twórczego, chociaż bywają chlubne wyjątki (np. w naszym kraju gdański oddział rozwojowy firmy Crosscom czy, także z Gdańska, Young Digital Poland) Zapewne nie zawsze tak być musi. Na targach CeBit97 na stoisku Islandii zdobyłem małą książeczkę (wydaną przez agencję rządową) na temat eksportu usług software'owych. Ten mały kraj (265 tysięcy mieszkańców) świadczy przez 50 małych domów software'owych (średnie zatr. 14 osób i obrot roczny ok. 1 mln dolarów) usługi dla Citybank, Visa Int., BP, Ford Motor Co, Kodak, ICAO, NATO itp. Zakres prac jest szeroki: od programów antywirusowych, oprogramowaniu bankowym typu point-of-sale, skończywszy na programach dla obrony powietrznej i projektowaniu hurtowni danych opartych na technologii OLAPu. Na tych samych targach, nasz kraj – dawny „dumny" producent systemu zarządzania bazą danych RODAN – zaprezentował się jako dostawca „podkładek" pod myszy.

Można zadać pytanie „czy motorem postępu jest krytyka?" i w zasadzie wypadałoby pozostawić je bez odpowiedzi. Rynki zbytu oprogramowania to są wielkie pieniądze (za niektóre systemy, np. bankowe płaci się kilkadziesiąt milionów dolarów) i jakiekolwiek akcenty krytyczne (nawet te konstruktywne) narażają Autorów wypowiedzi nawet na groźby procesów sądowych, represyjne zwolnienia z pracy itp.

### *Standardy de jure i de facto*

W warunkach działań „prawie monopolowych" w sposób naturalny powstają standardy – decyduje o nich często skala zastosowań, a nie standardy formalne opracowane przez międzynarodowe organizacje do tego celu powołane.

Od strony formalnej otwartość sprzętu i oprogramowania systemowo-narzędziowego (systemu operacyjnego, systemu zarządzania bazą danych, jezyków programowania itp.) określana jest jako Open Computing Systems (OCS). Otwartość od strony komunikacji i współdziałania komputerów określana jest przez standardy OSI (Open System Interconnection). Można więc powiedzieć, że spełnienie wymagań OCS i OSI daje nam otwartość platformy sprzętowo-systemowej.

Problemy otwartości systemów stanowią przedmiot działania kilku organizacji międzynarodowych, w tym stowarzyszenia X/Open (powołanego w połowie lat 80-tych), OSF (Open Software Foundation (1988), UI (Unix International) oraz OMG (Object Management Group). Należą do nich głównie producenci komputerów. W lutym 1996 r. w wyniku konsolidacji X/Open i OSF powstała organizacja The Open Group, która ma zespolić w całość standardy XWindow, Motif i CDE (Common Desk Environment), zaś głównym celem nadal pozostaje specyfikacja systemów otwartych.

Od strony standardów nieformalnych (de facto) obserwuje się ekspansję systemu operacyjnego Windows NT, a szczególnie w ostatnim okresie dotyczy to zastosowań bankowych na poziomie serwerów oddziałowych. W tym przypadku sprawa otwartości ulega pewnemu uproszczeniu, gdyż Microsoft nie jest satelitą jednego producenta komputerów i system ten od „urodzenia" jest wieloplatformowy, co nie oznacza, że nie posiada wad (np. brak standardowej obsługi katalogów globalnych co nieco obniża jego ocenę w aplikacjach klient-serwer ).

Architektura klient/serwer budowana powinna być wg standardów OSF – DCE (Open Software Foundation – Distributed Computing Environment) oraz OMG – CORBA (Object Management Group – Common Request Broker Architecture). Microsoft stosuje swój standard w tym zakresie pod nazwą DCOM (Distributed Object Model).

DCE narzuca m.in. algorytm generowania unikalnego identyfikatora UUID (Universal Unique Identifier), przeznaczonego do rozpoznawania użytkowników, grup, węzłów sieci, składnię zdalnego wywołania procedury (RPC – Remote Procedure Call) oraz standard obsługi katalogów globalnych i węzłowych (GDS – Global Directory Service, CDS – Cell Directory Service).

Standardy CORBA V.1.1.,v.1.2. i 2.0 dotyczą obiektowo zorientowanego standardu API w jednorodnym i heterogenicznym (zróżnicowanym) środowisku sprzętowo-systemowym.

## Co dalej?

**W niniejszym artykule – zgodnie zresztą z intencją i tytułem – poświęciłem dużo miejsca faktom historycznym. Aby chociaż nieco rozszerzyć obraz informatyki wymienię trendy charakterystyczne dla drugiej połowy lat 90-tych:**

- **architektura klient/server jako przeciwwaga do komputerów klasy mainframe,**
- **nowe wydajne minikomputery z równoległym przetwarzaniem,**
- **obiektowe podejście w projektowaniu i programowaniu oraz obiektowe bazy danych,**
- **wielowymiarowe hurtownie danych w skali korporacji i techniki eksploaracji danych,**
- **systemy ekspertowe i neuronowe, jako metoda rozwiązywania problemów nietypowych w warunkach niepełnych informacji i niepełnych algorytmów oraz narzędzie do szybkiego wskazywania stanów zagrożenia (włamań do sieci komputerowej, ryzyka kredytowego itp.),**
- **rozwój telekomunikacji, dzięki czemu stała się możliwa globalna łączność internetowa, globalne sieci komputerowe i bankowość elektroniczna.**

POLSKIE TOWARZYSTWO INFORMATYCZNE

# BIULETYN

NUMER 6 (135) ROK XVI CZERWIEC 1997

## Przyznano pierwsze w Polsce Europejskie Komputerowe Prawo Jazdy!

Uzyskał je Radosław Horodniczy, uczeń liceum ogólnokształcącego w Złotoryi. Pomyślnie zdał 7 egzaminów, które odbyły się we Wrocławiu. W dniu 17.IV.97 kandydat zdawał Moduły 1, 2 i 3, a 6.V.97 moduły 4, 5, 6 i 7. Otrzymał EKUK nr **PL-0400008.**

**Serdeczne gratulacje dla Radosława, Złotoryi i Wrocławia.**

### Nowi egzaminatorzy ECDL:

| | |
|---|---|
| Lech Pławecki | tel. (0-32) 1103648 |
| Janusz Trawka | tel. 0-602-304166, kiss@pik-net.pl |
| Dariusz Doliwa | tel. (0-42) 872614, doliwa@imul.uni.lodz.pl |
| Wojciech Horzelski | tel. (0-42) 524537, horzel@imul.uni.lodz.pl |
| Tomasz Gwiazda | tel. (0-22) 7511995, gwiazda@ocelot.wz.uw.edu.pl |
| Jolanta Rutkowska | tel. (0-22) 6386816, jolanta.@ocelot.wz.uw.edu.pl |
| Anna Cetnarowicz-Jutkiewicz | tel. (0-22) 6413128, nbpginbzi@telbank.pl |
| Teresa Tomaszewska | tel. (0-22) 6693372 |
| Alicja Myszor | tel. (0-58) 374358, amyszor@cto.gda.pl |
| Elżbieta Lipka | tel. (0-22) 6281460, lipka@cie.waw.pl |
| Jerzy Nowak | tel. (0-32) 382778, jnowak@silesia.pik-net.pl |
| Zenon Sosnowski | tel. (0-84) 617821, zenon@ai.pb.bialystok.pl |
| Andrzej Król | tel. (0-22) 658-43-68 |

## Rocznicowe wspomnienia o początkach PTI, Jerzym Trybulskim, konkursie Jego imienia i propozycji reaktywacji tego konkursu

*2[4] rocznica powstania PTI jest okazją, by poszperać nieco w archiwaliach Towarzystwa. Najlepszym źródłem jest oczywiście Biuletyn - prawdziwa skarbnica informatycznych wieści. Sięgnijmy więc do pierwszego numeru (z października 1982 roku), gdzie przypomniano okoliczności powstania PTI.*

*Sekretarz Generalny, Andrzej Wiśniewski pisał:*

W dniu 18 grudnia 1980 r. spotkała się w Pałacu Kultury i Nauki niewielka grupa osób zdecydowanych podjąć próbę stowarzyszenia polskich informatyków. Wybrany wówczas Komitet założycielski opracował i przedstawił w Urzędzie Miasta Stołecznego Warszawy zestaw dokumentów, m. in. projekt statutu, niezbędnych do zrealizowania tej inicjatywy. I oto **w dniu 22 maja 1981 roku** zostało zarejestrowane Polskie Towarzystwo Informatyczne, a dzień później odbył się Zjazd Założycielski. Podejmowane od kilkunastu lat próby stowarzyszenia informatyków zostały tym razem uwieńczone sukcesem...

*Uzupełnieniem niech będzie fragment tekstu zamieszczonego na stronie WWW naszego Towarzystwa:*

Członkowie Założyciele wywodzili się głównie ze środowisk naukowych uczelni i instytutów naukowych z całego niemal kraju. Szeregi Towarzystwa szybko się powiększały. Deklaracje członkowskie podpisało wielu informatyków zatrudnionych w przemyśle i urzędach. Rozproszeni po różnych ośrodkach informatycy w PTI znaleźli mocne oparcie, dzięki któremu mogli znacznie donośniej artykułować swoje racje i interesy zawodowe. Dzięki uruchomionej w połowie lat osiemdziesiątych działalności gospodarczej wielu członków miało szansę na realizację swoich ambicji zawodowych. Niewiele czasu minęło i obietnica pierwszego Prezesa, że „funduje prawdziwego francuskiego szampana dla całego Zarządu Głównego PTI, gdy na swym posiedzeniu przyjmie 1000 członka PTI" stała się faktem.

*A Pierwszym Prezesem Towarzystwa został wybrany prof. Władysław Turski, piastujący tę funkcję przez dwie kolejne kadencje. Oto jak określił dążenia Towarzystwa (Biuletyn nr 1):*

... Przekonani, że pełnoprawnym członkiem wspólnoty cywilizacyjnej nie może być kraj niezdolny do autonomicznego gromadzenia, przetwarzania i udostępniania informacji, zabiegamy o upowszechnienie zrozumienia konieczności objęcia państwowym mecenatem zastosowań informatyki...

...Dążymy do zintegrowania środowiska. W działalności Towarzystwa nie baczymy i nie chcemy baczyć na to, gdzie kto pracuje, lecz tylko na to, czy jest rzetelnym fachowcem informatykiem. Stanowisko takie przynosi niemałe dywidendy: szczerość wypowiedzi, zaufanie do ludzkiej solidności partnera...

... Krótko powiedziawszy, PTI zostało powołane z woli środowiska polskich informatyków w celu jednoczenia ich wysiłków na rzecz prawidłowego rozwoju polskiej informatyki, wyrażania uczciwych i kompetentnych opinii w tej sprawie, zgodnego ze współczesną wiedzą informowania społeczeństwa o informatyce i jej zastosowaniach, a także po to, by wspólnie i godnie dbać o dobre imię polskich informatyków. W wysiłkach na rzecz tych celów nie powinniśmy ustawać ani na chwilę.

*To przesłanie pozostało aktualne do dziś.*

*We władzach Towarzystwa zabrakło niestety **Jerzego Trybulskiego.** Wspomina prof. Zygmunt Mazur (Biuletyn PTI nr 9 z września 1985 r.):*

Na początku lat sześćdziesiątych istniała w środowisku wrocławskim zorganizowana grupa entuzjastów elektronicznej techniki obliczeniowej; przewodniczył jej Jerzy Trybulski. Z jej inicjatywy w 1964 r. utworzono pierwszy w Polsce, ogólnodostępny, usługowy ośrodek elektronicznej techniki obliczeniowej we Wrocławiu. Pionierską misję zorganizowania ośrodka powierzono Jerzemu Trybulskiemu. Przedsiębiorstwo to, prowadzone ofiarnym wysiłkiem garstki ludzi, zadziwiło swym rozwojem największych nawet optymistów. Czternastoletni okres działalności Jerzego Trybulskiego jako dyrektora ZETO we Wrocławiu, a następnie ponad dwuletni okres kierowania Stołecznym Ośrodkiem Informacji i Techniki Obliczeniowej - to niemal nieustanne pasmo osiąganych wielkim wysiłkiem sukcesów w dziedzinie programowania informatyki i pozyskiwania rynku, w realizacji śmiałych przedsięwzięć obejmujących coraz to nowe obszary zastosowań, implikujących rozwój bazy sprzętowej i technologii przetwarzania informacji. Jerzy Trybulski przykładał również ogromną wagę do spraw wychowania i kształcenia kadr informatycznych.

Jerzy Trybulski pełnił rolę Przewodniczącego Komisji Rejestracyjnej w Komitecie Założycielskim PTI. Nie było Mu jednak dane

wzięcie udziału w Zjeździe Założycielskim PTI. Choroba i jej tragiczny finał 23. 04. 1981 r., na miesiąc przed zarejestrowaniem Towarzystwa, zaskoczyły i pogrążyły w żalu środowisko informatyczne.

Osiągnięcia zawodowe zapewniają Jerzemu Trybulskiemu trwałe miejsce w historii polskiej informatyki. Dla upamiętnienia tego faktu - na mocy uchwały I Walnego Zjazdu PTI - Ogólnopolski Konkurs na najlepsze wdrożone prace z dziedziny zastosowań informatyki w gospodarce narodowej nazwano Jego imieniem. Organizację Konkursu powierzono Oddziałowi Dolnośląskiemu.

### *Konkurs imienia Trybulskiego*

Celem Konkursu była stymulacja szybszego rozwoju zastosowań informatyki w różnych dziedzinach gospodarki narodowej w drodze rozpowszechniania wyróżniających się rozwiązań informatycznych oraz stworzenia dodatkowych bodźców do podejmowania prac związanych z zastosowaniami informatyki.

W konkursie mogli wziąć udział obywatele Polski, autorzy prac wykonanych na terenie kraju i eksploatowanych przez użytkownika co najmniej przez jeden rok. Zgłoszone na konkurs prace miały być oceniane według kryterium wymiernych efektów ekonomicznych. Ustalono, że przedmiotem oceny nie będą wdrożenia dotyczące języków oprogramowania, systemów operacyjnych oraz szeroko rozumianej technologii komputerowego przetwarzania, oprogramowania narzędziowego i pomocniczego. Chodziło bowiem o popularyzację systemów, a nie narzędzi. W przemyśle wykorzystywane były przede wszystkim komputery RIAD, często projekty systemów wędrowały na półkę, nie dochodziło do wdrożeń. Na konkurs 1987 zgłoszono 7 wdrożonych systemów, w 1990 zgłoszono 5 systemów, w 1993 tylko 1 (!) system.

Za najlepsze wdrożone rozwiązanie informatyczne zgłoszone na konkurs przyznawano medal z wizerunkiem Trybulskiego. Komisja Konkursowa czcząc jego pamięć postanowiła pierwszy medal wraz z listem wręczyć Jego żonie, Pani Marii Trybulskiej.

### *Czy w dzisiejszych warunkach ekonomicznych uda się reaktywować ten konkurs?*

*Pomysł reaktywacji powstał w czerwcu 1996 roku w czasie V Walnego Zjazdu PTI. W uchwale zjazdowej ukazał się zapis zobowiązujący nowe władze do wznowienia Konkursu (wznowiony Biuletyn PTI nr 1, ze stycznia 1997 roku). Do sprawy powrócono we Wrocławiu w lutym 1997 roku, przy okazji rozstrzygnięcia konkursu na najlepsze prace magisterskie . Oddźwięk rozmów tam przeprowadzonych znalazł się na liście dyskusyjnej PTI-L. Rozpoczął Prezes, do dyskusji włączyli się Zygmunt Mazur i Jarek Deminet przypominając historię konkursów i ich uwarunkowania, a wspierający nasze działania z Toronto prof. Jacek Kryt przesłał wypowiedź na temat szans reaktwowania takiego Konkursu dzisiaj. Oto obszerne fragmenty tej wypowiedzi:*

... Systemy z okresu socjalizmu były w gospodarce sterowanej podaży. Drukowano mnóstwo raportów z wykonania planu o wartości archiwalnej. Czynniki produkcji dane do dyspozycji, określały wielkość produkcji. Zastosowanie miało programowanie liniowe, techniki badań operacyjnych czy ekonometrii. Opracowanie wymagań systemu nie było trudne. Można było łatwo modelować i programować bez troski o konsumenta. Nie interesowano się zbytem, jak również tym, kto system zakupi. Personel znał swe maszyny, narzędzia i jezyk programowania. Wymagania systemu docelowego były precyzyjne i bez niejasności. Budowa takich systemów nie wymagała zapewne więcej niż kilku miesięcy pracy. Łatwo było sprawdzić, czy system zadowoli te wymagania, czy więc był udany i powinien być nagrodzony.

W gospodarce rynkowej, nastawionej na zaspokajanie popytu, a więc na zmienny, trudny do prognozowania, dbający o opłacalność rynek, systemy pracują w orientacji na ruchomy cel, jakim jest prognoza popytu w zmieniającym się ciągle kontekście rynku (lokalnego i światowego). Dużo trudniej jest tu utrafić w dziesiątkę, budując dobry system na czas, w ramach określonego z góry budżetu, i robiący to, czego potrzebują użytkownicy. Budowanie większego systemu rozciąga się na miesiące i lata. Programiści i analitycy muszą rozumieć język i problemy użytkowników, a użytkownicy muszą wiedzieć, jak buduje się systemy skomputeryzowane i jaką oni sami muszą odegrać rolę, by system był dla nich pożyteczny. Użytkownicy w gospodarce rynkowej są faktycznymi pracodawcami informatyków. Mogą system przyjąć entuzjastycznie, albo odrzucić, zażądać poprawek lub uzupełnień, sabotować, i nie używać. Kryteria sukcesu są bez porównania trudniejsze do określenia niż w gospodarce sterowanej. To bardzo wiele kryteriów do obiektywnego osądu, czy uwzględnienia w decyzji o wartości wdrożonego systemu informacyjnego w „rynkowej" gospodarce narodowej. To wymaga też dłuższej obserwacji zachowania się systemu.

Podaję kilka przykładów nadawania nagród przez Canadian Information Processing Society (CIPS), Kanada.

1. Nagroda za Kanadyjskie Oprogramowanie Narzędziowe,
2. Nagroda za Innowacje w Technologii Informacji, (dotyczy zarówno innowacji handlowych jak i niehandlowych)
3. Nagroda za Obsługę Klientów Systemów (Customer Service),
4. Nagrody za produktywność w dziedzinie informacji w Kanadzie (ustanowiona w 1996 roku, za koordynację „czynnika ludzkiego", procesu i technologii),
5. Nagroda C.C.Gotlieb za działania na rzecz CIPS,
6. GALA (Got A Lot Accomplished) - Za wielkie osiągnięcie! nadawana osobom, które walnie przyczyniły się do promocji CIPS'u w skali krajowej,
7. Nagroda za osiągnięcia zawodowe
8. Członkostwo honorowe CIPS'u

*Czekamy na Państwa głosy w tej sprawie.*

## Czerwcowe kalendarium OddziaŁu Górnośląskiego

- 5-6 czerwca - II Ogólnopolska Konferencja „ Systemy Informatyczne w przemyśle, II Targi MRP", Warszawa, PKiN
- 16-19 czerwca, IX Wyższa Górska Szkoła PTI „Modelowanie i organizacja przepływu prac" - Ustroń

## Wiosenny Konkurs Programowania Zespołowego w Poznaniu

Jesienią czeka nas II Ogólnopolski Konkurs w Programowaniu Zespołowym. Myśląc o nim Oddział Wielkopolski PTI, Koło Naukowe Informatyków BooBoo, Instytut Informatyki i sponsor Dantek Poznań zjednoczyły się, aby młodym ludziom kilka orzechów do zgryzienia przygotować. W sobotę 26.05 łamano więc głowy aby:

- znaleźć wspólny podobszar dwu prostokątnych okien wyświetlonych na ekranie (7:1),
- obliczyć najmłodszą, niezerową cyfrę n! (7:1)
- sprawdzić, czy składniki i wynik sumy lub różnicy nie przekraczają zakresu liczb całkowitych (6:2),
- zbadać spójność grafu i ustalić dla tego grafu liczbę różnych dróg zawierających 2 wierzchołki i 3 wierzchołki (3:5),
- wypisać słownie podaną kwotę złotych (5:3).

(w nawiasach wynik meczu rozwiązane : nierozwiązane)

Pierwsze trzy zespoły zniszczyły wszystkie zadania. Aby jesienią było podobnie!

*Jan Kniat*

## TO NIE SEN - TO JAVA

Ponownie zapraszamy do udziału w II Letniej Szkole Programowania Wizualnego Poznań '97, która odbędzie się w terminie **9-11.09.1997**. Zgłoszenia pisemne można nadsyłać do **10.07.1997.** Szczegóły znajdują się na stronie WWW:

http://www.cs.put.poznan.pl/~java. Kontakt: java@cs.put.poznan.pl.

Redaktor: EWA ŁUKASIK
e-mail: lukasik@pozn1v.put.poznan.pl, tel. (61) 782 373,
Instytut Informatyki, Politechnika Poznańska, ul. Piotrowo 3a, 60-965 Poznań

Polskie Towarzystwo Informatyczne, Zarząd Główny
tel. (22) 624 60 61 w.328, tel./fax (22) 652 32 59
ul. Żelazna 87, 00-879 Warszawa, URL: http://www.pol.pl/pti

I N T E R N E T KONCESJA 29/96/J

File Edit Window Help

BACK NEXT STOP HOME LOOK BEST CODE PRINT HELP

Site: http://heros.juni.com/

http://heros.juni.gliwice.pl./

# 1987 – 1997

## JUNISOFTEX OFERUJE:

- Nowoczesne obiektowe aplikacje
- Zarządzanie procesem projektowym wsparte narzędziami CASE (w tym Select OMT)
- Analizy dziedzinowe oparte o metodę łączącą nowoczesność z 20 letnim doświadczeniem wdrożeniowym
- Sprawne i szybkie wdrożenia zgodne z obowiązującymi przepisami
- Projektowanie i wykonawstwo sieci komputerowych
- Typowe, publiczne usługi internetowe dla osób prawnych i fizycznych
- Technologie internetu do restrukturyzacji procesów organizacyjnych
- Technologie internetu w systemach wspomagających proces zarządzania
- Integracje różnych rodzajów sieci
- dostawy sprzętu komputerowego renomowanych firm
- dostawy sprzętu peryferyjnego

# Perfect-Ekspert '97

**System Perfect-Ekspert'97 jest zbiorem aplikacji integrujących procesy gospodarcze w całym przedsiębiorstwie.**

**Strategia zarządzania i systemy informowania kierownictwa**
Sprawozdawczość Obserwacja Dociekanie Wnioskowanie Komentowanie Symulacje i Prognozy

**System finansowo-kosztowy**
Banki Kasa Sprzedaż Zakup Koszty proste Rozliczanie kosztów Sprawozdawczość

**System ewidencji majątku**
Ewidencja Inwentaryzacja Koszty Plan amortyzacji Przeszacowania Sprawozdawczość

**Systemy obsługi klienta, obrót towarowy, sprzedaż**
Zamówienia Ewidencja Inwentaryzacja Zakup towarów Kompletacja Ewidencja opakowań zwrotnych Sprzedaż Analizy Sprawozdawczość

**Systemy gospodarki materiałowej i zaopatrzenia**
Zakupy Ewidencja Inwentaryzacja Dyspozycja Rezerwacja Koszty Analiza Sprawozdawczość

**System kadrowo-płacowy oraz dyscypliny**
Dokumentacja i ewidencja Sytemy wynagrodzeń Czas pracy i rejestratory czasu pracy Kwalifikacje i szkolenia Emisja dokumentacji ZUS i zasiłki Sprawozdawczość

**System przygotowania produkcji**
Dokumentacja technologiczna Narzędzia i urządzenia pomiarowe Kalkulacje i ofertowanie Emisja dokumentacji warsztatowej Plan zakupów Plany warsztatowe Produkcja w toku Rozliczenie produkcji

**Systemy pomiarowe i stanowiska pomiarowe**
Statystyczne oceny jakości Dokumentacja jakości Świadectwa jakości i atesty Baza pomiarowa

Przedsiębiorstwo Produkcyjno–Usługowe Innowacji Technicznych

**JUNISOFTEX® Sp. z o.o.**

ul. Kościuszki 1c, 44-100 GLIWICE
tel.: (0-32) 130 69 31 do 33
fax: (0-32) 130 69 34
E-mail: DYREKTOR@JUNIGLIWICE.PL
E-mail: PREZES@JUNIGLIWICE.PL
Nr Polpak: 13211054
NIP: 631-020-03-11
Regon: P-008012971-94927177-59-3-671-27177
Konto: PKO I/O Gliwice 27661-225298-136
Dyrektor Naczelny – Krzysztof Niewiadomski
Prezes – Bronisław Jureczko
Kapitał zakładowy 1.150.000.000 PLZ (115.000 PLN)

Firma JUNISOFTEX obecna jest na rynku od 10 lat. W tym czasie wdrożono kilkaset dużych aplikacji. Naszej firmie zaufało jak do tej pory około 100 przedsiębiorstw produkcyjnych w tym 6 spółek giełdowych, 10 przedsiębiorstw NFI, kilka przedsiębiorstw przemysłu hutniczego i wiele innych.

## Lista aktualnych użytkowników „Perfect-Ekspert"

- Elektrownia RYBNIK w Wielopolu
  jedyna w Polsce elektrownia rozliczająca koszty rzeczywiste remontów na 3 tys. obiektów
- ELEKTROBUDOWA S.A. w Katowicach
- POLIFARB S.A. w Cieszynie
- WEDEL w Warszawie
- P.P.O.W. AGROS-PIŃCZÓW w Pińczowie
- Zakłady Odzieżowe BYTOM S.A. w Bytomiu
- Maszyny Elektryczne CELMA w Cieszynie
- Fabryka Amortyzatorów KROSNO S.A. w Krośnie
- Instytut Spawalnictwa w Gliwicach
- GEOFIZYKA w Toruniu
- POLSPORT w Bielsku-Białej
- POLAM-KOSTUCHNA w Katowicach
- Walcownia Metali DZIEDZICE S.A. w Czechowicach-Dziedzicach
- ELEKTRONARZĘDZIA CELMA S.A. w Cieszynie
- Zakłady Tworzyw Sztucznych KRYWAŁD ERG w Knurowie
- Fabryka Aparatów Elektrycznych APENA S.A. w Bielsku-Białej
- MOSTOSTAL Słupca
- Fabryka Osi Napędowych w Radomsku
- Huta Szkła Okiennego KUNICE S.A. w Żarach
- Huta ŁAZISKA w Łaziskach Górnych
- ELEKTROMONTAŻ 3 S.A. w Katowicach
- Fabryka Palenisk Mechanicznych w Mikołowie
- DRAGMOR sp. z o.o. w Szczecinie
- Dąbrowskie Zakłady Naprawcze Przemysłu Węglowego w Dąbrowie Górniczej
- Centrum Naukowo Techniczne Kolejnictwa w Warszawie
- Zakłady Rowerowe ROMET Kowalewo
- SUWARY S.A. w Pabianicach
- Zakłady Naprawcze Przemysłu Węglowego w Rybniku
- Rejonowe Przedsiębiorstwo Wodociągów i Kanalizacji w Gliwicach
- Zakłady Tłuszczowe BIELMAR S.A. w Bielsku-Białej
- GZUT w Gliwicach
- MEFTA w Mikołowie
- KRONOPOL w Żarach
- Huta BANKOWA
- Walcownia Metali ŁABĘDY S.A. w Gliwicach
- Huta 1-go MAJA w Gliwicach
- ELEKTROMETAL Zakład Telemechaniki Górniczej w Cieszynie
- Zakłady Metalurgiczne TRZEBINIA w Trzebini
- Huta SILESIA w Rybniku
- Fabryka Automatyki Chłodniczej S.A. w Cieszynie
- Zakłady Górniczo-Hutnicze BOLESŁAW w Bukownie
- MIFAMA w Mikołowie

Document: Loading

Koncepcja i pomysł: K. Niewiadomski, wykonanie: „F-11"

# URODZENI ZWYCIĘZCY

## Najwyższej klasy komputery HP dostępne w sieci dealerów ABC DATA.

Do każdego zakupionego komputera dołączamy bezpłatnie pakiet Lotus SmartSuite 96

The Intel Inside Logo and Pentium are registered trademarks and MMX™ is a trademark of the Intel Corporation. All other trademarks are acknowledged.

ABC DATA CHS EXCELLENCE IN DISTRIBUTION

Warszawa, tel. 633 70 11
Gdynia, tel. 29 75 95
Kraków, tel. 21 46 56
Poznań, tel. 76 48 91

APRIL BUSINESS COMPUTER
(22) 633 42 85

ASUB
(22) 649 45 61

COMPUTERLAND
(22) 634 56 66

DYNAMIC SOFTWARE
(22) 612 48 64

FLOPPY
(56) 54 26 67

FONTEX
(22) 620 41 16

K.O.T.
(22) 827 30 62

MEDICAT-SYSTEM
(22) 831 44 70

MICOMP
(32) 51 30 86

NET TECHNOLOGY
(22) 652 21 31

OFFICE DEPOT
(22) 720 46 76

ORION INSTUMENTS
(22) 644 36 97

PC DIRECT
(22) 675 29 40

RIMCO
(22) 617 32 91

SEKOM
(22) 675 71 74

SIECIOWE SYSTEMY INFORMACYJNE
(32) 51 70 77

SMK
(22) 632 20 31

STER PROJEKT
(22) 43 14 11

SURFLAND
(71) 73 56 13

TWINS
(22) 625 46 32

UNIA
(22) 606 02 78

Komputery Hewlett-Packard od lat zwyciężają w fachowych zestawieniach. Ale nie to jest najważniejsze. Ich moc i niezawodność pomaga zwyciężać tysiącom firm, w których są niezastąpionym narzędziem zwiększania wydajności pracy. Dzień po dniu, rok po roku, pracują na sukces swoich użytkowników. Perfekcyjnie.

Przychodzi im to łatwo - są do tego stworzone. Wyposażyliśmy je w nowatorskie funkcje ułatwiające obsługę i podnoszące bezpieczeństwo, a wszystkie zastosowane rozwiązania przeszły złożone, wszechstronne testy na stanowiskach pracy.

Profesjonalny i szybki serwis zapewniamy tylko na wszelki wypadek.

Dlatego komputery HP zwyciężają na co dzień.

Dalsze informacje uzyskasz z serwera: http://www.hp.com lub w Dziale Obsługi Klienta HP, tel. (22) 608 74 03.

**Vectra VE**

• Procesor Intel Pentium® 133/166 MHz • 8 lub 16 MB EDO RAM, rozszerzalne do 192 MB • 256 KB cache synchr. pipe-line burst (opcja) • 1,0 lub 1,6 GB HDD • 64 bitowy akcelerator graficzny PCI • HP Dynamic Video • Kontroler Enhanced IDE PCI • 4 wolne sloty (2 PCI, 1 Combo, 1 ISA) • Software DMI • Zarządzanie poborem mocy VEPS • Włączanie z klawiatury, funkcja HP-off dla Windows 95 • Zaawansowane funkcje bezpieczeństwa • Windows 95 • 3 lata gwarancji

**Vectra VL**

• Procesor Intel Pentium® z technologią MMX™ 133/200 MHz, upgrade • 8/16/32 MB EDO RAM, rozszerzalne do 192 MB • 256 lub 512 KB cache synchr. pipe-line burst • 1,6 lub 2,5 GB HDD • 64 bitowy akcelerator graficzny PCI • Kontroler Enhanced IDE PCI • w modelach multimedialnych: CD-ROM IDE 8x oraz Sound Blaster™ Pro • 4 lub 6 wolnych slotów • Software DMI • Zarządzanie poborem mocy VEPS • Włączanie z klawiatury, funkcja HP-off dla Windows 95 • Zaawansowane funkcje bezpieczeństwa • Windows 95 • 3 lata gwarancji

**Vectra XA**

• Procesor Intel Pentium® 133 MHz, Intel Pentium® z technologią MMX™ 166/200 MHz, Intel Pentium® Pro 180/200 MHz • 16/32 MB EDO RAM, rozszerzalne do 192 MB • 256 KB cache synchr. pipe-line burst lub zintegrowana z procesorem• Napęd HDD 1,6/2,5 GB Enhanced IDE PCI lub 2,1/4,5 GB Ultra SCSI PCI • akcelerator graficzny Matrox Millenium • w większości modeli: CD-ROM IDE 8x oraz Sound Blaster™ Pro • 4 lub 6 wolnych slotów • Software DMI • Włączanie z klawiatury, funkcja HP-off dla Windows 95 • Zaawansowane funkcje bezpieczeństwa i zarządzania poborem mocy • Windows 95 lub Windows NT 4,0 • 3 lata gwarancji

**NetServer E 30**

• Procesor Intel Pentium® 166 MHz • 16 MB ECC RAM (z detekcją i korygowaniem błędów), rozszerzalne do 192 MB • 256 cache synchr. pipe-line burst • HDD 2,1 GB Fast SCSI-2 • Kontroler Fast SCSI-2 PCI • akcelerator graficzny PCI • CD-ROM IDE 4x • 6 wolnych slotów (2 PCI, 2 Combo, 2 ISA) • 6 półek na pamięci masowe • Software DMI • Zarządzanie poborem mocy VEPS • ASR - automatyczne restartowanie serwera • Zaawansowane funkcje bezpieczeństwa • 3 lata gwarancji.

intel inside
pentium® PROCESSOR

HEWLETT® PACKARD

FRANCE 98 WORLD CUP

OFICJALNY DOSTAWCA SPRZĘTU